

UNIVERSITÉ DE SHERBROOKE
Faculté de génie
Département de génie électrique et de génie informatique

RÉDUCTEURS DE BRUIT ADAPTATIFS SPATIAUX ET
POST-TRAITEMENT POUR CODEC MPEG-2

Thèse de maîtrise es sciences appliquées
Spécialité : génie électrique

DO VIET Ha

Sherbrooke (Québec), Canada

Hiver 2000



National Library
of Canada

Acquisitions and
Bibliographic Services

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque nationale
du Canada

Acquisitions et
services bibliographiques

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence

Our file Notre référence

The author has granted a non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of this thesis in microform, paper or electronic formats.

The author retains ownership of the copyright in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de cette thèse sous la forme de microfiche/film, de reproduction sur papier ou sur format électronique.

L'auteur conserve la propriété du droit d'auteur qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

0-612-67258-1

Canada

À toute ma famille et à tous mes amis

SOMMAIRE

Jusqu'à présent, plusieurs techniques adaptatives spatiales pour la réduction de bruit additif ont été développées. Mais chaque algorithme peut seulement être efficace pour une classe d'images et il dépend beaucoup du niveau de bruit incorporé. Dans ce travail, nous proposons de nouvelles approches qui permettent d'obtenir une meilleure performance pour toutes les images et pour tous les niveaux de bruit. Les images traitées par nos algorithmes ont une meilleure qualité non seulement en mesure objective mais aussi subjective. Les algorithmes proposés sont fondés sur les statistiques locales des images, de l'énergie de bruit et de la segmentation. Grâce à la segmentation qui distingue les zones plates et les zones de détails dans l'image à traiter, l'utilisation du filtre adaptatif spatial devient plus précise et plus efficace, et introduit un meilleur résultat comparativement aux filtres existants.

Puisque nos réducteurs de bruit sont capables d'enlever même le bruit dans les zones autour des contours, nous les utilisons pour développer un système de post-traitement pour améliorer la qualité visuelle des images vidéo compressées par le standard MPEG-2. Dans de telles régions, le bruit de moustique nuit beaucoup à la qualité visuelle. Plus spécifiquement, dans les images de MPEG-2, il y a trois types de bruits principaux : bruit de moustique, bruit de quantification et l'effet de bloc. Notre système commence par un filtre en losange, qui permet d'enlever une certaine quantité de bruit de quantification et d'effet de bloc. Ensuite, pour réduire le bruit de moustique, ce qui est très important, nous utilisons nos réducteurs de bruit. Enfin, un traitement de quatre sous-bandes est développé pour réduire l'effet de bloc.

REMERCIEMENTS

Je tiens à exprimer ma gratitude en remerciant chaleureusement tous ceux et celles qui m'ont aidé et encouragé durant toutes mes études. En particulier, mon directeur de recherche LE DINH Chon Tam, pour ses conseils judicieux, sa disponibilité et son soutien financier.

Je tiens aussi à remercier Miranda Technologies Inc. pour le financement du projet.

J'exprime ma profonde reconnaissance à ma femme, mes parents, frères, sœurs et amis pour l'encouragement et le soutien accordé tout au long de ce programme.

TABLE DES MATIÈRES

INTRODUCTION.....	1
RÉDUCTEURS DE BRUIT ADDITIF	5
2.1 RÉDUCTEURS DE BRUIT EXISTANTS.....	5
2.1.1 <i>Algorithme utilisant les statistiques locales</i>	10
2.1.1.1 Modèle de l'image	10
2.1.1.2 Filtrage optimal	10
2.1.2 <i>Algorithme utilisant le filtre Sigma</i>	14
2.1.3 <i>Algorithme utilisant quatre filtres unidimensionnels</i>	16
2.2 RÉDUCTEURS DE BRUIT PROPOSÉS	21
2.2.1 <i>Algorithme de segmentation</i>	21
2.2.2 <i>Algorithmes dérivés de l'approche de segmentation</i>	25
2.2.2.1 Approche utilisant la moyenne des images	26
2.2.2.2 Approche utilisant une image constante.....	27
2.2.2.3 Approche combinée	29
2.3 COMPARAISON DE LA PERFORMANCE ENTRE LES RÉDUCTEURS DE BRUIT	31
2.3.1 <i>Le rapport signal sur bruit</i>	33
2.3.2 <i>L'efficacité de la réduction de bruit</i>	34
2.3.3 <i>La conservation des contours et des détails fins</i>	39
2.3.4 <i>La complexité</i>	39
2.3.5 <i>Sommaire</i>	40
POST-TRAITEMENT POUR CODEC MPEG-2.....	49
3.1 CODEC MPEG-2	50
3.1.1 <i>Types d'images</i>	51
3.1.2 <i>Hiérarchie dans le système de codage</i>	52
3.1.3 <i>Techniques de compression MPEG-2</i>	54
3.1.3.1 Réduction des redondances temporelles	54
3.1.3.2 Réduction des redondances spatiales	56
3.1.3.3 Système d'encodage.....	61
3.1.3.4 Système de décodage	63
3.2 ARTEFACTS CAUSÉS PAR LE CODEC MPEG-2.....	65
3.2.1 <i>Effet de bloc</i>	65
3.2.2 <i>Bruit de moustique</i>	66
3.2.3 <i>Bruit de quantification</i>	67
3.3 POST-TRAITEMENT.....	67
3.3.1 <i>Traitement de la luminance Y</i>	68
3.3.1.1 Estimation de la puissance de bruit.....	70
3.3.1.2 Réduction du bruit de moustique et du bruit aléatoire	71
3.3.1.3 Réduction des effets de bloc	73
3.3.2 <i>Traitement des chrominances U et V</i>	76
3.3.3 <i>Présentation des résultats</i>	78
CONCLUSION	92
BIBLIOGRAPHIE.....	95

LISTE DES FIGURES

Figure 2.1 : Modèle d'une image dégradée par un bruit additif.....	6
Figure 2.2 : Exemples du filtre passe-bas spatial.....	6
Figure 2.3 : Réduction de bruit par des filtres moyens.....	7
Figure 2.4 : Système adaptatif typique pour la réduction de bruit.....	8
Figure 2.5 : Système utilisant les statistiques locales.....	13
Figure 2.6 : Algorithme utilisant le filtre Sigma.....	16
Figure 2.7 : Système en cascade de quatre filtres unidimensionnels.....	17
Figure 2.8 : Algorithme de segmentation.....	23
Figure 2.9 : Segmentation en utilisant la filtre passe-bas et la valeur de seuil.....	24
Figure 2.10 : Algorithme utilisant la moyenne des images.....	26
Figure 2.11 : Algorithme utilisant une image constante.....	28
Figure 2.12 : Algorithme combiné.....	31
Figure 2.13 : Comparaison des rapports signal sur bruit.....	36
Figure 2.14 : Comparaison des rapports signal sur bruit.....	37
Figure 2.15 : Comparaison des rapports signal sur bruit entre les trois algorithmes.....	38
Figure 2.16 : Image « Airplane » dégradée par le bruit gaussien additif.....	41
Figure 2.17 : Résultats de l'image « Airplane » ($\sigma^2=400$) traitée par les algorithmes.....	42
Figure 2.18 : Image « Lenna » dégradée par le bruit gaussien additif.....	43
Figure 2.19 : Résultats de l'image « Lenna » ($\sigma^2=400$) traitée par les algorithmes.....	44
Figure 2.20 : Image « Couple » dégradée par le bruit gaussien additif.....	45
Figure 2.21 : Résultats de l'image « Couple » ($\sigma_n^2=400$) traitée par les algorithmes.....	46
Figure 2.22 : Image « Bridge » dégradée par le bruit gaussien additif.....	47
Figure 2.23 : Résultats de l'image « Bridge » ($\sigma_n^2=400$) traitée par les algorithmes.....	48
Figure 3.1 : Signal vidéo entrelacé et progressif.....	50
Figure 3.2 : I-, P-, et B-images dans une séquence de vidéo MPEG-2.....	52
Figure 3.3 : Hiérarchie dans le système de codage.....	53
Figure 3.4 : Compensation de mouvement bidirectionnelle.....	55
Figure 3.5 : Transformation des pixels en coefficients DCT.....	57
Figure 3.6 : Parcours en zigzag.....	60
Figure 3.7 : Schéma-bloc de l'encodeur.....	61
Figure 3.8 : Arrangement des images dans la séquence pour la compensation de mouvement.....	62
Figure 3.9 : Schéma-bloc du décodeur.....	64
Figure 3.10 : Image décodée par MPEG-2 à un débit de 3Mbit/s.....	66
Figure 3.11 : Système du post-traitement de MPEG-2 pour la luminance Y.....	68
Figure 3.12 : Réponse fréquentielle du filtre en losange à 75%.....	69
Figure 3.13 : Système d'estimation du bruit.....	70
Figure 3.14 : Courbe d'estimation d'écart-type du bruit.....	71
Figure 3.15 : Systèmes de réduction du bruit de moustique et du bruit aléatoire.....	72
Figure 3.16 : Système de réduction des effets de bloc.....	74
Figure 3.17 : Localiser les pixels à corriger dans les directions verticales, horizontales et diagonales.....	75
Figure 3.18 : Sous-échantillonnage horizontal de la luminance Y.....	77
Figure 3.19 : Réponse fréquentielle du filtre passe-bas $F(\omega)$	77
Figure 3.20 : Rapport signal sur bruit des séquences vidéo décodées.....	80
Figure 3.21 : Rapport signal sur bruit du post-traitement pour la séquence « Mobile ».....	81
Figure 3.22 : Rapport signal sur bruit du post-traitement pour la séquence « Mobile ».....	82
Figure 3.23 : Résultats du post-traitement de la luminance Y pour « Mobile » à 3Mbit/s.....	83
Figure 3.24 : Résultats du post-traitement des chrominance U et V pour « Mobile » à 3 Mbit/s.....	86
Figure 3.25 : Résultats du post-traitement de la luminance Y pour « Flower » à 3 Mbit/s.....	88
Figure 3.26 : Résultats du post-traitement des chrominance U et V pour « Flower » à 3 Mbit/s.....	90

LISTE DES TABLEAUX

<i>Tableau 2.1 : Rapports signal sur bruit des images dégradées en dB.....</i>	<i>34</i>
<i>Tableau 2.2 : Complexité des algorithmes.....</i>	<i>40</i>
<i>Tableau 3.1 : Codes de longueur variable pour les coefficients AC quantifiés.....</i>	<i>60</i>

Chapitre 1

Introduction

Le rehaussement et la restauration des images dégradées sont toujours deux problèmes importants dans le traitement des images. En général, le rehaussement vise à modifier l'apparence d'une image de manière à ce qu'un observateur ou une machine puisse extraire plus facilement une certaine information désirée. La restauration, quant à elle, a pour objectif de rétablir la qualité d'une image dégradée pour qu'elle soit la plus proche de celle de l'image originale [11,16]. Cependant, le rehaussement et la restauration d'images sont reliés de très près. Si une image est dégradée, sa restauration donne normalement un bon résultat dans le rehaussement. Le développement d'un système de restauration dépend du type de la dégradation. Il est clair qu'un algorithme destiné à enlever le bruit gaussien additif est différent de celui qui est utilisé pour réduire le brouillage dans les images [16]. Les algorithmes discutés dans ce projet sont conçus essentiellement pour la réduction de bruit dans les images.

La réduction de bruit est un domaine relié aux techniques d'amélioration de la qualité des images contaminées par le bruit. Ce domaine empiète aussi sur la restauration des images. Durant les dernières années, plusieurs techniques de réduction de bruit ont été développées. La différence entre elles dépend du type de bruit des

images. La performance des algorithmes se modifie en fonction des facteurs suivants [9,11] :

- l'objectif du traitement;
- le critère utilisé pour juger la qualité de l'image traitée;
- le modèle statistique du bruit;
- le type et la nature de l'image à traiter.

En général, les techniques de réduction de bruit peuvent être réparties en deux grandes catégories : les méthodes spatiales et les méthodes fréquentielles. Les premières sont fondées sur la manipulation directe des pixels dans une image. La plupart de ces approches font appel à l'application des filtres moyens et des filtres médians qui, habituellement, servent à enlever le bruit de l'image sans besoin d'une identification explicite du bruit [6,7,26]. Les méthodes fondées sur les fréquences ont pour objet de modifier la transformation de Fourier d'une image. On peut réduire le bruit grâce à la conception des filtres fréquentiels et à l'adaptation d'une fréquence de coupure quand les composantes du bruit ne sont pas en corrélation avec le signal utile dans le domaine fréquentiel [5]. Récemment, plusieurs approches fondées sur la transformation en ondelettes ont été proposées [1,4,19]. Selon ces méthodes, l'image contaminée est d'abord transformée en coefficients d'ondelettes. Des coefficients supposés dégradés par le bruit sont ensuite remplacés par zéro ou par une autre valeur appropriée. Les coefficients qui restent peuvent être manipulés aussi. Enfin, l'image traitée est obtenue par synthèse à partir des coefficients d'ondelettes.

Dans ce travail, nous utilisons les méthodes spatiales de réduction de bruit. On peut trouver dans cette catégorie différentes techniques de filtrage. Mais il est difficile de déterminer quelle est la meilleure technique de filtrage. En général, plus le filtre peut s'adapter aux caractéristiques locales de l'image, plus le résultat est bon [23]. Dans cette recherche, nous avons un objectif principal et un secondaire. Le premier consiste à concevoir un nouveau réducteur de bruit adaptatif qui permet une meilleure

performance que les algorithmes précédents, et à faire une comparaison détaillée de ces performances. Comme objectif secondaire, nous utilisons ce nouveau réducteur de bruit pour développer un système de post-traitement du signal vidéo après encodage/décodage à l'aide de la norme MPEG-2.

MPEG-2 est une technique standard pour la compression des images vidéo appliquée à la distribution de la télévision, par exemple. Ce standard permet de compresser des images par un rapport 40 à 1 ou plus [2]. Cependant, quand le facteur de compression est élevé, l'algorithme commence à générer des défauts préjudiciables à la qualité de perception des images. La dégradation des images devient particulièrement importante si les séquences vidéo contiennent beaucoup de mouvements et beaucoup de détails fins.

En fait, il y a trois principaux types de dégradations sur les images décodées par le standard MPEG-2. La première est bien connue sous le nom de « effet de bloc ». Cette erreur se manifeste sous la forme d'une grande variation des valeurs des pixels aux frontières de blocs voisins. En regardant une image complète, on peut remarquer qu'il y a des zones constituées de blocs indépendants. La deuxième dégradation porte le nom de « bruit de moustique ». En général, cette erreur se trouve dans les zones plates contenant des contours de faible amplitude. Dans une séquence d'images animées, on peut apercevoir des effets bizarres ressemblant à des moustiques et qui clignent autour des contours. La dernière dégradation est le bruit de quantification. Ce bruit apparaît quand les niveaux de quantification deviennent grands pour parvenir à un plus haut taux de compression. En effet, ce bruit est un bruit aléatoire de nature [11,25].

Le post-traitement que nous proposons a pour objectif de réduire les bruits mentionnés ci-haut pour obtenir des séquences vidéo de meilleure qualité pour la perception humaine, non seulement dans la luminance mais aussi dans la chrominance des images.

Cet ouvrage débute au chapitre 2 par la présentation des réducteurs dans le cas du bruit aléatoire additif. Dans la première section, nous présentons les études portant sur les réducteurs de bruit adaptatifs existants. Bien qu'il existe plusieurs techniques possibles, nous nous concentrons sur les trois approches principalement utilisées dans les systèmes de réduction de bruit. Il s'agit de la technique des statistiques locales [13], du filtre Sigma [14] et de la mise en cascade de quatre filtres unidimensionnels adaptatifs [3]. Nous proposons, par la suite, nos réducteurs de bruit. Enfin, la troisième section est consacrée à la comparaison de la performance de ces réducteurs à celle des autres techniques existantes.

Ces réducteurs de bruit sont ensuite appliqués au post-traitement du signal vidéo MPEG-2, qui fait l'objet du troisième du chapitre 3. Dans la première section de ce chapitre, nous présentons les notions de base du codec MPEG-2. Par la suite, nous abordons les défauts causés par le codec MPEG-2. La dernière section est consacrée à la présentation du système proposé pour améliorer la qualité visuelle des images vidéo MPEG-2.

Chapitre 2

Réducteurs de bruit additif

Ce chapitre se compose de trois sections. La première section présente une recherche bibliographique sur les techniques adaptatives spatiales de réduction de bruit additif. À partir de ce travail, nous choisissons et décrivons trois approches principales qui permettent généralement d'obtenir une bonne performance. Ces méthodes sont celle exploitant les statistiques locales [13], celle qui utilise le filtre Sigma [14], et l'algorithme de Chan et Lim utilisant la cascade de quatre filtres unidimensionnels [3]. Dans la deuxième section, nous proposons de nouveaux réducteurs de bruit. La troisième section est consacrée à la comparaison des résultats obtenus avec ces réducteurs par rapport aux approches existantes. Cette comparaison est effectuée à l'aide d'images de natures différentes, dégradées par des niveaux de bruit variables.

2.1 Réducteurs de bruit existants

Dans le domaine de réduction de bruit, il existe plusieurs systèmes pour traiter les différents types de bruit. Généralement, les bruits que l'on rencontre sont le bruit

additif, le bruit multiplicatif et le bruit impulsionnel ou un mélange des trois. À fin de mieux réduire ces bruits, il faut concevoir pour chacun un modèle différent.

Normalement, le modèle d'une image dégradée par un bruit additif est représenté par l'équation 2.1 et par la figure 2.1 :

$$g(n_1, n_2) = f(n_1, n_2) + n(n_1, n_2) \quad (2.1)$$

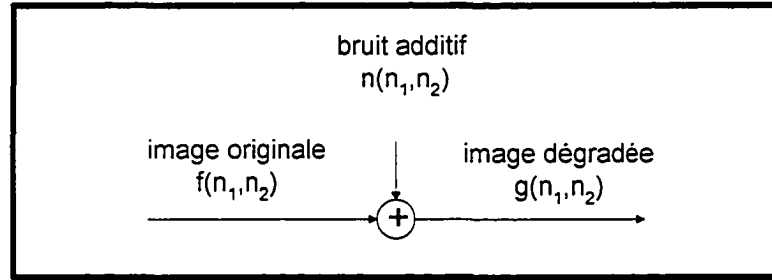


Figure 2.1 :Modèle d'une image dégradée par un bruit additif

où $n(n_1, n_2)$ représente le bruit additif indépendant du signal de l'image. Le bruit dans les circuits électroniques est un exemple du bruit gaussien additif [11].

$$h_1(n_1, n_2) = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} / 9 \quad h_2(n_1, n_2) = \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} / 16 \quad h_3(n_1, n_2) = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix} / 25$$

Figure 2.2 :Exemples du filtre passe-bas spatial.

Une des méthodes simples pour réduire ce bruit est le filtre passe-bas spatial. En passant l'image dégradée par ce filtre, on obtient une image beaucoup plus lisse. Par contre, ce filtre fait disparaître beaucoup de détails utiles de l'image, notamment quand la dimension du filtre devient considérable. La figure 2.2 donne quelques exemples de ces filtres et la figure 2.3 présente les résultats après l'application du filtre $h_1(n_1, n_2)$ et $h_3(n_1, n_2)$ sur l'image « Lenna », dégradée par un bruit aléatoire additif de moyenne nulle et d'une variance de 144.



Figure 2.3 : Réduction de bruit par des filtres moyens

- a) Image originale; b) Image dégradée;
 c) Image traitée par le filtre $h_1(n_1, n_2)$; d) Image traitée par le filtre $h_3(n_1, n_2)$

Dans ces filtres, les coefficients sont invariants et appliqués à l'image entière, tandis que les caractéristiques de l'image se changent considérablement d'une région à l'autre [11]. Par exemple, les régions du fond de la fille ont des intensités approximativement uniformes alors que les détails sur son chapeau présentent une grande variation locale de l'intensité. Ceci explique pourquoi l'utilisation de ces filtres

brouille de vrais contours, et enlève beaucoup de détails à l'image. Alors, un système de réduction du bruit qui s'adapte aux caractéristiques de l'image et du bruit serait plus efficace.

Durant les dernières années, plusieurs techniques adaptatives ont été développées dans le but de restaurer les images dégradées par les bruits. La plupart des algorithmes adaptatifs spatiaux de ces travaux peuvent être présentés dans le système représenté par la figure 2.4 [16].

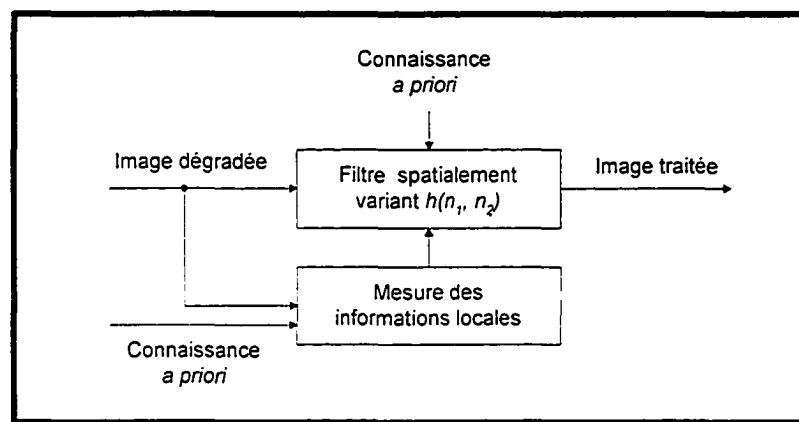


Figure 2.4 : Système adaptatif typique pour la réduction de bruit

Dans ce système, à partir de l'image dégradée et de la connaissance *a priori*, on peut déterminer certaines mesures des caractéristiques locales de l'image sans bruit, à savoir la variance locale et la moyenne locale. Appliqué sur une image dégradée, le filtre $h(n_1, n_2)$ dans la figure 2.4 est de type spatialement variant en fonction des caractéristiques locales de l'image et des connaissances *a priori*. En effet, il est un filtre passe-bas quand le bruit dans l'image a une grande largeur de bande. Dans les zones à basse fréquence comme celles à l'intensité constante, où le bruit est plus visible que dans les zones à plusieurs détails, on peut mettre en œuvre une grande partie du filtrage passe-bas pour réduire le bruit le mieux possible. Cette opération n'a pas beaucoup d'effet sur les composantes du signal parce que la variation du signal utile dans les zones plates est très petite. Par contre, dans les zones à haute fréquence, où se trouvent

les détails de l'image comme des contours, seulement un faible filtrage est exécutée. Celui-ci ne peut pas réduire beaucoup de bruit, mais il conserve les composantes du signal. En plus, dans les zones à plusieurs détails, le bruit ne peut pas être réduit facilement comme dans les zones plates, si l'on désire conserver les détails pertinents [11,16].

En exploitant ces principes de base, un nombre important d'algorithmes ont été développés : les filtres médians uni et bidimensionnels [1,4-7,19,23,26], la méthode de réduction de bruit de Nagao et Matsuyama [21], les techniques utilisant le «filtre boîte » et le filtre Sigma [14,20], la méthode exploitant les statistiques locales [13] et l'approche de Chan et Lim utilisant la cascade de quatre filtres unidimensionnels [3]. La différence de ces algorithmes réside dans les facteurs suivants :

- les mesures spécifiques utilisées pour représenter les caractéristiques locales de l'image;
- la façon de déterminer le filtre spatialement variant $h(n_1, n_2)$;
- les connaissances *a priori* disponibles.

En fait, il est difficile de classer ces filtres selon leur importance parce que l'un peut être efficace pour une classe d'images donnée mais il ne l'est pas pour une autre. Dans les sections suivantes, nous étudions trois algorithmes principaux développés par Lee, Chan et Lim [3,13,14]. Ceux-ci sont généralement efficaces et permettent une bonne performance dans la réduction du bruit additif des images dégradées.

2.1.1 Algorithme utilisant les statistiques locales

Dans cet algorithme, aussi bien que dans ceux présentés dans les prochaines sections, nous supposons que le bruit additif $n(n_1, n_2)$ au pixel de coordonnées (n_1, n_2) est centré et blanc avec une moyenne nulle et une variance σ_n^2 connue d'avance. C'est une hypothèse qui s'avère réaliste, parce que les caractéristiques du bruit sont souvent bien connues. Même dans les cas où elles ne sont pas disponibles, on peut les estimer en utilisant les techniques d'apprentissage de données ou à partir de l'image dégradée donnée [14,24]. Selon l'hypothèse de base, la moyenne et la variance de chaque pixel de l'image sont égales respectivement à la moyenne locale et à la variance locale de l'ensemble des pixels contenus dans une fenêtre centrée sur le pixel d'intérêt.

2.1.1.1 Modèle de l'image

Considérons une petite région dans laquelle le signal $f(n_1, n_2)$ de l'image originale est supposé stationnaire. Dans cette région locale, il peut être modélisé par l'équation [16] :

$$f(n_1, n_2) = m_f + \sigma_f w(n_1, n_2) \quad (2.2)$$

où m_f et σ_f sont la moyenne et l'écart-type locaux de l'image $f(n_1, n_2)$; $w(n_1, n_2)$ est un bruit blanc, de moyenne nulle et de variance unitaire. Selon l'équation 2.2, l'image $f(n_1, n_2)$ est modélisée par la somme de la moyenne locale m_f et d'un bruit blanc de variance locale σ_f^2 qui sont tous les deux spatialement variants.

2.1.1.2 Filtrage optimal

Le filtre minimisant l'erreur quadratique moyenne entre l'image originale $f(n_1, n_2)$ et l'image dégradée $g(n_1, n_2)$ possède la réponse en fréquence suivante [16] :

$$H(\omega_1, \omega_2) = \frac{P_f(\omega_1, \omega_2)}{P_f(\omega_1, \omega_2) + P_n(\omega_1, \omega_2)} = \frac{\sigma_f^2}{\sigma_f^2 + \sigma_n^2} \quad (2.3)$$

où (ω_1, ω_2) désignent les fréquences horizontales et verticales et

$P_f(\omega_1, \omega_2)$ = densité spectrale de la puissance du signal $f = \sigma_f^2$

$P_n(\omega_1, \omega_2)$ = densité spectrale de puissance du bruit $n = \sigma_n^2$

De l'équation 2.3, on déduit la réponse impulsionnelle du filtre $h(n_1, n_2)$ par la transformation de Fourier inverse :

$$h(n_1, n_2) = \frac{\sigma_f^2}{\sigma_f^2 + \sigma_n^2} \delta(n_1, n_2) \quad (2.4)$$

En appliquant l'équation 2.4, l'image traitée $p(n_1, n_2)$ dans une région locale peut être calculée par l'équation suivante [16] :

$$\begin{aligned} p(n_1, n_2) &= m_f + (g(n_1, n_2) - m_f) \left(\frac{\sigma_f^2}{\sigma_f^2 + \sigma_n^2} \right) \delta(n_1, n_2) \\ &= m_f + \frac{\sigma_f^2}{\sigma_f^2 + \sigma_n^2} (g(n_1, n_2) - m_f) \end{aligned} \quad (2.5)$$

Si on suppose que la moyenne locale m_f et la variance locale σ_f^2 sont modifiées à chacun pixel de l'image, on obtient cette équation :

$$p(n_1, n_2) = m_f(n_1, n_2) + \frac{\sigma_f^2(n_1, n_2)}{\sigma_f^2(n_1, n_2) + \sigma_n^2} (g(n_1, n_2) - m_f(n_1, n_2)) \quad (2.6)$$

L'algorithme basé sur cette équation a été développé par Lee J.S [13]. Il peut être considéré comme un cas spécial du processus à deux canaux selon lequel l'image à traiter est divisée en deux composantes, la moyenne locale $m_f(n_1, n_2)$ et le contraste local $g(n_1, n_2) - m_f(n_1, n_2)$. Ces deux facteurs peuvent être modifiés séparément et les

résultats obtenus seront combinés ensemble. Dans l'approche de Lee (L'équation 2.6), la moyenne locale $m_f(n_1, n_2)$ n'est pas modifiée, tandis que le contraste local est multiplié par un facteur. Si σ_f^2 est beaucoup plus grand que σ_n^2 , le niveau de détails de l'image est important et le contraste local de l'image dégradée $g(n_1, n_2)$ n'est pas atténué. Dans ce cas, $p(n_1, n_2)$ est approximativement égal à $g(n_1, n_2)$. Si σ_f^2 est beaucoup plus petit que σ_n^2 , le niveau de bruit est important et le contraste local de $g(n_1, n_2)$ est significativement atténué. Dans de telles régions, l'image traitée $p(n_1, n_2)$ est approximativement égale à $m_f(n_1, n_2)$ et devient très lisse.

Si la moyenne m_n du bruit est égale à zéro, la moyenne m_f devient identique à la moyenne m_g et on peut estimer $m_f(n_1, n_2)$ dans l'équation 2.6 de l'image dégradée $g(n_1, n_2)$ en utilisant l'équation ci-dessous :

$$\hat{m}_f(n_1, n_2) = \hat{m}_g(n_1, n_2) = \frac{1}{(2p+1)(2q+1)} \sum_{k_1=n_1-p}^{n_1+p} \sum_{k_2=n_2-q}^{n_2+q} g(k_1, k_2) \quad (2.7)$$

où $(2p+1)(2q+1)$ est le nombre de pixels dans une fenêtre utilisée pour l'estimation. À l'intérieur de cette fenêtre, la variance locale $\sigma_f(n_1, n_2)$ peut être supposée spatialement invariante. De même, $\sigma_f(n_1, n_2)$ peut être estimée à partir de l'image dégradée $g(n_1, n_2)$ par l'équation suivante :

$$\hat{\sigma}_f^2(n_1, n_2) = \begin{cases} \hat{\sigma}_g^2(n_1, n_2) - \sigma_n^2 & \text{pour } \hat{\sigma}_g^2(n_1, n_2) > \sigma_n^2 \\ 0 & \text{ailleurs} \end{cases} \quad (2.8)$$

où $\hat{\sigma}_g^2(n_1, n_2)$ est calculée ainsi :

$$\hat{\sigma}_g^2(n_1, n_2) = \frac{1}{(2p+1)(2q+1)} \sum_{k_1=n_1-p}^{n_1+p} \sum_{k_2=n_2-q}^{n_2+q} (g(k_1, k_2) - \hat{m}_g(n_1, n_2))^2 \quad (2.9)$$

Si on définit le gain $a(n_1, n_2)$ par

$$a(n_1, n_2) = \begin{cases} \frac{\hat{\sigma}_g^2(n_1, n_2) - \sigma_n^2}{\hat{\sigma}_g^2(n_1, n_2)} & \text{pour } \hat{\sigma}_g^2(n_1, n_2) > \sigma_n^2 \\ \sigma_g^2(n_1, n_2) & \text{ailleurs} \\ 0 & \end{cases} \quad (2.10)$$

cet algorithme de Lee pour la réduction de bruit peut être illustré par la figure ci-dessous [11]:

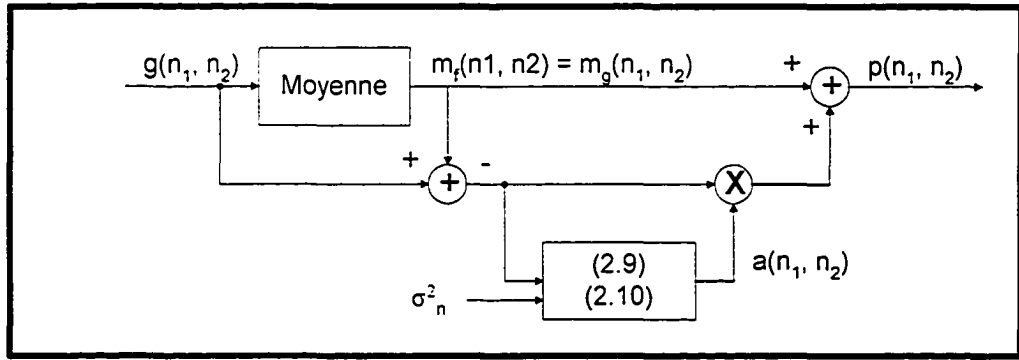


Figure 2.5 : Système utilisant les statistiques locales

Puisque cette technique est temporellement non-réursive, elle peut être appliquée dans les systèmes qui fonctionnent en temps réel.

En fait, cet algorithme s'avère efficace sur une classe d'images qui contient beaucoup de zones plates. Par contre, dans les zones de contours et de détails fins, il ne réduit pas beaucoup le bruit, et dans le cas où le rapport signal sur bruit est faible, ce bruit devient visible pour la perception humaine. Dans la prochaine section, nous étudions une autre technique qui permet de réduire le bruit, même dans les zones de contour. Il s'agit de l'algorithme utilisant le filtre Sigma.

2.1.2 Algorithme utilisant le filtre Sigma

Dans la section précédente nous avons discuté de l'approche de Lee, qui s'adapte aux caractéristiques locales de l'image. À l'intérieur d'une région, l'image est typiquement supposée stationnaire. Mais dans les zones où il y a des contours ou des lignes, cette hypothèse ne convient pas autant. Le filtre basé sur cette hypothèse préserve des contours, mais il laisse aussi une quantité significative de bruit autour de ces contours.

Une autre approche qui permet de surmonter efficacement ce problème a été développée aussi par Lee en 1983 [14]. Cet algorithme est motivé par l'écart-type d'une distribution aléatoire. Il utilise le principe du filtre moyen qui a pour objectif de remplacer le pixel central à traiter par la moyenne des pixels d'une région locale. La différence réside dans le fait que seuls les pixels dont l'intensité se trouve à l'intérieur de l'intervalle $[-2\sigma; +2\sigma]$ autour du pixel central sont pris en compte pour calculer la moyenne. C'est ce qu'on appelle la moyenne de deux-sigma. Avec un bruit gaussien de moyenne nulle, 95.5% des valeurs aléatoires se trouvent dans l'intervalle $[-2\sigma; +2\sigma]$. Des pixels à l'extérieur de cet intervalle, viennent souvent d'une autre population et doivent être exclus de la moyenne. En général, le filtre Sigma peut être conçu selon les étapes suivantes :

- a) Établir un intervalle des intensités $(g(n_1, n_2) - \Delta, g(n_1, n_2) + \Delta)$ où $\Delta = 2\sigma$.
- b) Sélectionner tous les pixels à l'intérieur d'une fenêtre de dimension $(2p+1)(2q+1)$ centrée autour du pixel d'intérêt et dont l'intensité appartient à l'intervalle $[-\Delta; +\Delta]$.
- c) Calculer la moyenne des pixels sélectionnés $m_f(n_1, n_2)$.
- d) Attribuer la valeur de la moyenne $m_f(n_1, n_2)$ à $p(n_1, n_2)$.

D'une façon mathématique, l'image traitée $p(n_1, n_2)$ peut être obtenue par l'équation 2.11 :

$$p(n_1, n_2) = \frac{\sum_{k_1=n_1-p}^{n_1+p} \sum_{k_2=n_2-q}^{n_2+q} g(k_1, k_2) w(k_1, k_2)}{\sum_{k_1=n_1-p}^{n_1+p} \sum_{k_2=n_2-q}^{n_2+q} w(k_1, k_2)} \quad (2.11)$$

où

$$w(k_1, k_2) = \begin{cases} 1 & \text{si } (g(n_1, n_2) - \Delta \leq g(k_1, k_2) \leq g(n_1, n_2) + \Delta) \\ 0 & \text{ailleurs} \end{cases} \quad (2.12)$$

Cette approche possède plusieurs avantages. Tout d'abord, non seulement le bruit dans les zones plates mais aussi le bruit autour des contours sont totalement enlevés. Ensuite, dans certains cas, cette réduction ne brouille pas les contours et les lignes dans l'image parce que seuls les pixels des contours sont inclus dans le calcul de la moyenne. De plus, cet algorithme ne produit pas des artefacts secondaires et garde les formes originales de l'image parce qu'il n'y a aucun traitement directionnel ici. La complexité d'une telle technique présente également un avantage : la vitesse de calcul est très rapide, l'algorithme se limitant à de simples comparaisons et additions.

Son grand désavantage est relié au bruit de tache : ce phénomène se produit lorsque le moyennage s'effectue sur un ensemble restreint à 1 ou 2 pixels. Cet effet devient particulièrement néfaste quand l'image est dégradée par un bruit de forte amplitude. À fin de remédier à ce problème, on remplace la moyenne de deux-sigma par celle des pixels voisins immédiats du pixels central dans le cas où le nombre de pixels M à l'intérieur de l'intervalle $[-\Delta; +\Delta]$ est plus petit qu'une valeur précise K . L'étape d) mentionnée précédemment doit donc être modifiée de la façon suivante :

$$p(n_1, n_2) = \begin{cases} m_f(n_1, n_2), \text{ moyenne de deux - sigma} & \text{pour } M > K \\ m_{\text{voisin}}(n_1, n_2), \text{ moyenne des pixels voisins} & \text{pour } M \leq K \end{cases} \quad (2.13)$$

Alors, la technique du filtrage de deux-sigma pour réduire de bruit gaussien additif peut être représentée par le système ci-dessous :

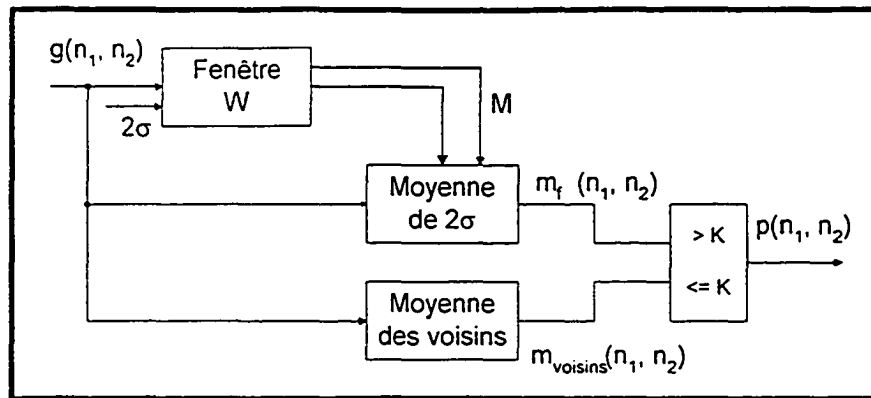


Figure 2.6 : Algorithme utilisant le filtre Sigma

La valeur K doit être choisie attentivement pour réduire le bruit de tache, sans enlever les détails fins de l'image. En réalité, cette valeur dépend de la dimension de la fenêtre de traitement. Pour une fenêtre de 7×7 , K doit être plus petite que quatre et pour une fenêtre de 5×5 , elle doit être plus petite que trois. Malgré cette précaution, une partie des détails fins disparaît en même temps que le bruit de tache. Plus concrètement, tous les détails fins des textures, qui se trouvent dans l'intervalle d'intensités de deux-sigma, sont aussi enlevés à l'image. Il s'agit là d'un vrai problème causé par cet algorithme. Dans les cas où l'image à traiter contient beaucoup de détails fins, on perdra une quantité importante d'informations.

2.1.3 Algorithme utilisant quatre filtres unidimensionnels

Une autre technique, qui permet une bonne performance dans certains cas, a été développée par Chan et Lim [3]. Cet algorithme utilise une cascade de quatre filtres adaptatifs unidimensionnels orientés dans quatre directions principales de l'image : horizontale, verticale, diagonale de 45 degrés et diagonale de 135 degrés. Chaque filtre utilise le même modèle de l'image et les mêmes principes de base que le développement de l'algorithme 2-D présenté dans la section 2.1.1, sauf qu'il traite l'image comme un signal unidimensionnel. Dans cette technique, on vise à réduire le bruit dans toutes les régions de l'image tout en conservant ses contours. Le principe de

cet algorithme est simple : si un filtre est orienté dans la même direction que le contour, celui-ci sera préservé tandis que le bruit de l'image sera filtré selon cette direction, inversement si le filtre n'est pas dans la même direction que le contour l'écart-type local sera grand et le contour sera conservé.

On présente $H_i[.]$ comme un filtre unidimensionnel, obtenu de la même façon que celui dans l'approche 2-D, mais qui se base sur une région locale 1-D orientée dans la $i^{ième}$ direction. Les directions sont normalement orientées selon quatre angles principaux de 0° , 45° , 90° et 135° . L'image dégradée $g(n_1, n_2)$ est filtrée par ces filtres unidimensionnels en cascade en supposant que l'image est un signal unidimensionnel pour chaque filtre. L'image traitée $p(n_1, n_2)$ peut être obtenue mathématiquement par l'équation et la figure suivantes :

$$p(n_1, n_2) = H_4[H_3[H_2[H_1[g(n_1, n_2)]]]] \quad (2.14)$$

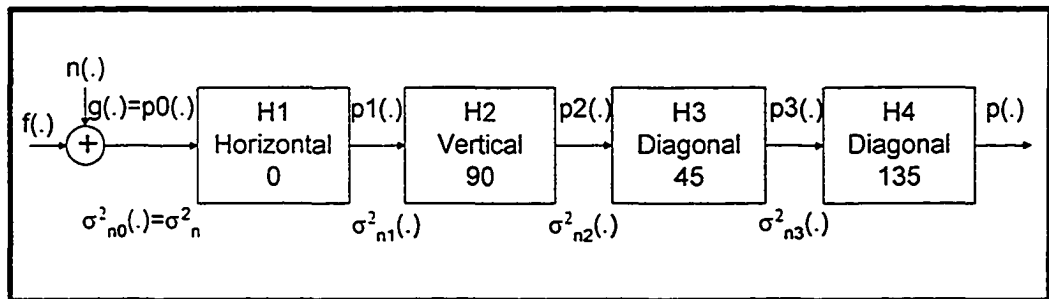


Figure 2.7 : Système en cascade de quatre filtres unidimensionnels

Comme le filtre bidimensionnel discuté dans la section 2.1.1, les filtres $H_i[.]$ sont aussi spatialement variants. En regardant l'équation 2.14, il est bien clair que l'image dégradée $g(n_1, n_2)$ est filtrée séquentiellement. Pour cette raison, les caractéristiques locales du bruit changent après le traitement de chaque filtre. Elles doivent donc être recalculées après l'application de chaque filtre $H_i[.]$.

Prenons l'exemple du filtre unidimensionnel dans la direction horizontale. Si l'on utilise la même méthode que celle de l'approche bidimensionnelle présentée dans la section 2.1.1, on obtient les équations suivantes :

$$p_1(n_1, n_2) = \hat{m}_f(n_1, n_2) + \frac{\sigma_f^2(n_1, n_2)}{\sigma_f^2(n_1, n_2) + \sigma_n^2(n_1, n_2)} (g(n_1, n_2) - \hat{m}_f(n_1, n_2)) \quad (2.15)$$

où

$$\hat{m}_f(n_1, n_2) = \hat{m}_g(n_1, n_2) = \frac{1}{(2p+1)} \sum_{k_1=n_1-p}^{n_1+p} g(k_1, n_2) \quad (2.16)$$

$$\hat{\sigma}_f^2(n_1, n_2) = \begin{cases} \hat{\sigma}_g^2(n_1, n_2) - \sigma_n^2(n_1, n_2) & \text{pour } \hat{\sigma}_g^2(n_1, n_2) > \sigma_n^2(n_1, n_2) \\ 0 & \text{ailleurs} \end{cases} \quad (2.17)$$

avec

$$\hat{\sigma}_g^2(n_1, n_2) = \frac{1}{(2p+1)} \sum_{k_1=n_1-p}^{n_1+p} (g(k_1, n_2) - \hat{m}_g(n_1, n_2))^2 \quad (2.18)$$

Dans ces équations, $p_1(n_1, n_2)$ représente la sortie du premier filtre. Si le deuxième filtre unidimensionnel est vertical, et en utilisant la même méthode, on obtiendra l'image de sortie $p_2(n_1, n_2)$ avec l'entrée $p_1(n_1, n_2)$. Pour les deux autres directions diagonales, les filtres sont appliqués de façon similaire. Comme nous l'avons mentionné, le niveau de bruit dans l'image diminue à chaque sortie. Alors, dans cet algorithme, l'élément le plus important est de recalculer la variance du bruit, qui est réduite après chaque étape de filtrage. Ce calcul est assez compliqué mais il peut être effectué en utilisant les informations des filtres unidimensionnels et la variance du bruit donnée à l'avance.

Pour faciliter la compréhension du calcul de la variance du bruit, nous commençons avec un filtre unidimensionnel horizontal d'une longueur de 5. Tout d'abord, il faut trouver la réponse impulsionnelle de ce filtre. De l'équation 2.16, on constate que l'image des moyennes locales $m_f(n_1, n_2)$ est obtenue par la convolution de

l'image dégradée $g(n_1, n_2)$ et d'un filtre unidimensionnel invariant $f_1(k)$. On obtient alors :

$$\hat{m}_f(n_1, n_2) = g(n_1, n_2) * f_1(k; n_1, n_2) \quad (2.19)$$

avec

$$f_1(k; n_1, n_2) = \left[\frac{1}{5}, \frac{1}{5}, \frac{1}{5}, \frac{1}{5}, \frac{1}{5} \right] \quad (2.20)$$

Si l'on définit $d(n_1, n_2)$ comme la différence entre l'image dégradée $g(n_1, n_2)$ et celle de la moyenne locale $\hat{m}_f(n_1, n_2)$, elle peut être produite de façon similaire en passant l'image $g(n_1, n_2)$ par le filtre $f_2(k)$ avec les coefficients qui suivent :

$$f_2(k; n_1, n_2) = \delta(k; n_1, n_2) - f_1(k; n_1, n_2) = \left[-\frac{1}{5}, -\frac{1}{5}, \frac{4}{5}, -\frac{1}{5}, -\frac{1}{5} \right] \quad (2.21)$$

De l'équation 2.15, on sait bien que l'image de sortie $p_1(n_1, n_2)$ est calculée par la somme de l'image de la moyenne locale $m_f(n_1, n_2)$ et du produit de l'image de différence $d(n_1, n_2)$ et le facteur $a(n_1, n_2)$ recalculé à chaque pixel.

$$p_1(n_1, n_2) = \hat{m}_f(n_1, n_2) + a(n_1, n_2)d(n_1, n_2) \quad (2.22)$$

où

$$a(n_1, n_2) = \frac{\sigma_f^2(n_1, n_2)}{\sigma_f^2(n_1, n_2) + \sigma_n^2(n_1, n_2)} \quad (2.23)$$

En appliquant les équations 2.19, 2.20, 2.21, 2.22 et 2.23, on obtient la réponse impulsionnelle du filtre unidimensionnel horizontal $H_1[.]$.

$$h_1(k; n_1, n_2) = \left[\frac{1-a(.)}{5}, \frac{1-a(.)}{5}, \frac{1+4a(.)}{5}, \frac{1-a(.)}{5}, \frac{1-a(.)}{5} \right] \quad (2.24)$$

Dans cette approche, on suppose que l'image et le bruit sont indépendants. La variance du bruit après le filtre horizontal est calculée par l'équation suivante :

$$\sigma_{n1}^2(n_1, n_2) = \sigma_{n0}^2(n_1, n_2) \left(\frac{1 + 4a(.)}{5} \right)^2 + \quad (2.25)$$

$$\left[\sigma_{n0}^2(n_1 - 2, n_2) + \sigma_{n0}^2(n_1 - 1, n_2) + \sigma_{n0}^2(n_1 + 1, n_2) + \sigma_{n0}^2(n_1 + 2, n_2) \right] \left(\frac{1 - a(.)}{5} \right)^2$$

D'une façon générale, avec un filtre unidimensionnel horizontal de longueur $(2p+1)$ la variance du bruit modifiée est recalculée par la formule ci-dessous:

$$\sigma_{n1}^2(n_1, n_2) = \sum_{k=n1-p}^{n1+p} h_1^2(k; n_1, n_2) \sigma_{n0}^2(k, n_2) \quad (2.26)$$

où la réponse impulsionnelle du filtre est déterminée par cette équation :

$$h_1(k; n_1, n_2) = \left[\frac{1 - a(.)}{(2p+1)}, \dots, \frac{1 - a(.)}{(2p+1)}, \frac{1 + 2pa(.)}{(2p+1)}, \frac{1 - a(.)}{(2p+1)}, \dots, \frac{1 - a(.)}{(2p+1)} \right] \quad (2.27)$$

Pour les autres étapes du filtrage, les calculs de la variance du bruit et des coefficients des filtres unidimensionnels sont effectués de façon similaire.

Cet algorithme de Chan et Lim s'adapte aux contours dirigés suivant une des directions principales. Pour les images ayant cette caractéristique, l'algorithme fonctionne très bien; il réduit le bruit à proximité des contours et des lignes tout en préservant ces caractéristiques de l'image. Cependant, dans les cas où les contours de l'image ont une direction différente des quatre directions principales, ce système les réduit aussi bien que le bruit, ce qui fait perdre beaucoup de détails de l'image traitée. De plus, dans les zones plates, la cascade des quatre filtres unidimensionnels de longueur $(2p+1)$ est équivalent à un filtre bidirectionnel de dimensions $(6p+1) \times (6p+1)$. Celui-ci enlève tous les détails fins de la texture de l'image quand le niveau de bruit de l'image à traiter est faible. Il est efficace et donne une image lisse pour la perception humaine dans le cas où ce niveau de bruit ajouté est élevé.

Dans les prochaines sections, nous proposons d'autres méthodes qui peuvent combiner les algorithmes de Lee et de Chan et Lim pour une meilleure performance.

2.2 Réducteurs de bruit proposés

Dans les sections précédentes, nous avons étudié les deux algorithmes de Lee, et celui de Chan et Lim qui sont efficaces dans une certaine mesure et sont utilisés dans plusieurs systèmes de réduction de bruit des images.

Cependant, ces algorithmes dépendent beaucoup de la nature, de la structure et du rapport signal sur bruit des images à traiter. La technique utilisant les statistiques locales fonctionne bien quand les images contiennent beaucoup des zones plates et que le rapport signal sur bruit de l'image dégradée est grand. Par contre, quand le niveau de bruit ajouté est élevé, une quantité visible de bruit restant autour des contours qui se trouvent dans les zones plates de l'image. L'approche du filtre Sigma est meilleure dans certains cas, mais, en général, on perd de détails fins sur la texture de l'image. L'algorithme de Chan et Lim peut résoudre le problème qui subsiste avec l'approche utilisant les caractéristiques locales, mais il dépend aussi de la nature de l'image. De plus, l'estimation de la variance du bruit résiduel après filtrage de l'équation 2.25 établit l'hypothèse que le signal d'image et le bruit sont indépendants. Cette supposition est seulement valide pour la première étape du filtrage. Pour les autres étapes, dans les directions verticales et diagonales, le signal d'image et le bruit deviennent corrélés. De plus, à cause du traitement directionnel, cette technique produit des faux contours, ce qui deviennent particulièrement néfastes quand le niveau de bruit ajouté est élevé.

2.2.1 Algorithme de segmentation

Une approche qui peut surmonter ces difficultés consiste à modéliser l'image de façon plus précise et à développer un système de réduction de bruit en se basant sur ce nouveau modèle. Cependant, il est difficile de concevoir un nouveau modèle d'image, surtout qu'un algorithme de restauration fondé sur un modèle précis et détaillé est, en général, extrêmement complexe et requiert beaucoup de calcul. Une autre approche

consiste tout d'abord à extraire les contours de l'image à traiter à l'aide d'algorithmes spécifiques. En suite, un filtre adaptatif est développé en tenant compte de la carte des contours. Cependant, la complexité de cet algorithme est aussi grande que l'approche précédente et la détection de contours de l'image en présence du bruit reste une tâche très difficile [3,26].

En général, plus une technique de réduction de bruit s'adapte aux caractéristiques locales de l'image à traiter, plus le résultat est bon [23]. Dans cette section, nous présentons une autre approche proposée par LeDinh Chon-Tam* qui utilise le même modèle d'image que celui utilisé ci-avant mais permet une meilleure performance que les algorithmes existants. Typiquement, toutes les images contiennent des zones plates et des zones de contours et de détails. Les structures de ces zones influencent beaucoup la performance des techniques de réduction de bruit. Un algorithme performant sur un large éventail d'images doit être capable de distinguer les zones plates et les zones de contours dans une région donnée.

Cette nouvelle technique exploite aussi le principe de base de l'approche utilisant les caractéristiques locales de Lee présentée dans la section 2.1.1, toutefois avec une amélioration. L'amélioration principale de cet algorithme est la segmentation de l'image afin de distinguer où sont les zones plates et où sont les zones de détails et de contours. Grâce à cette segmentation, l'utilisation du filtre spatialement variant devient plus précise et plus efficace.

* Communication verbale

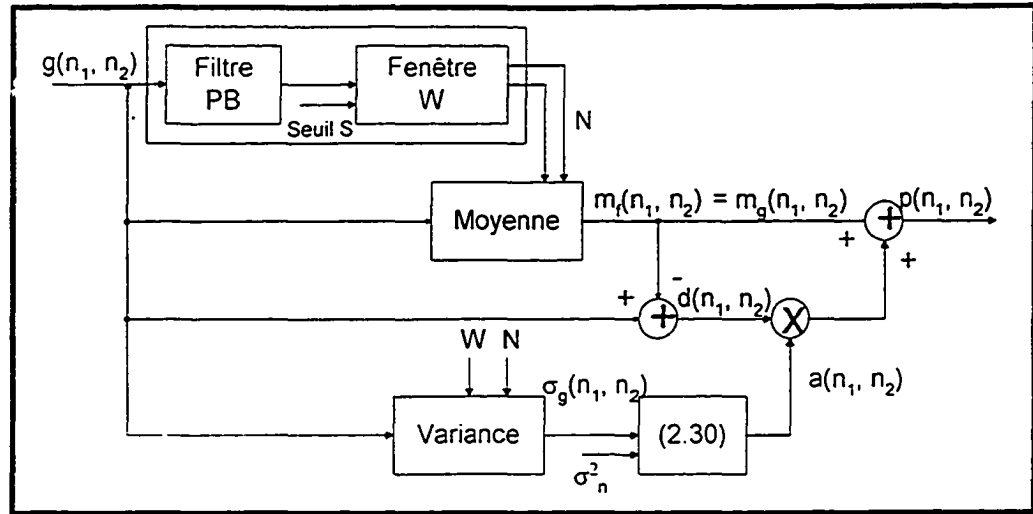


Figure 2.8 : Algorithme de segmentation

La figure 2.8 représente le système de réduction de bruit utilisant la segmentation. Au lieu d'utiliser tous les pixels à l'intérieur de la fenêtre de traitement pour calculer la moyenne locale $m_f(n_1, n_2)$ et la variance locale $\sigma_f^2(n_1, n_2)$, ce système commence par déterminer à quelle région le pixel appartient, et considère seulement les pixels dont l'intensité est similaire à celle du pixel central de la fenêtre de traitement. Dans l'exemple présenté dans la figure 2.9 ci-dessous, le nombre des pixels sélectionnés est égal à quatorze.

Cette segmentation est exécutée en appliquant un filtre passe-bas et une valeur de seuil précise S . Puisque le bruit dans l'image dégradée représente normalement les composantes à haute fréquence, l'énergie de l'image se trouve principalement concentrée à basse fréquence [11]. L'utilisation du filtre passe-bas vise à trouver les vraies régions de détails et de contours de l'image originale. À l'intérieur de la fenêtre de traitement, tous les pixels dont la valeur absolue de leur différence par rapport au pixel central est plus petite que la valeur de seuil S , se trouvent dans la même région, et ils sont pris en compte dans les calculs. Le filtre passe-bas peut être un filtre de dimension 3×3 ou 5×5 dont les coefficients sont les mêmes que ceux des filtres $h_1(n_1, n_2)$ et $h_3(n_1, n_2)$ présentés dans la figure 2.2 de la section 2.1.

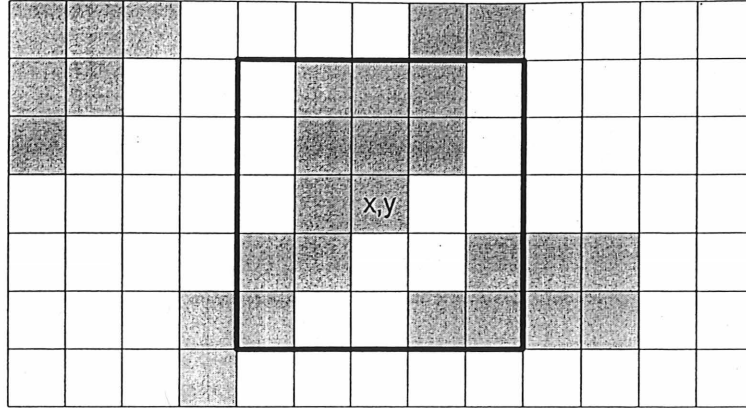


Figure 2.9 : Segmentation en utilisant la filtre passe-bas et la valeur de seuil

Après la segmentation, en appliquant la même méthode que celle de l'approche utilisant les caractéristiques locales de Lee présentée dans la section 2.1.1, on obtient les équations suivantes :

$$p(n_1, n_2) = \hat{m}_f(n_1, n_2) + a(n_1, n_2)(g(n_1, n_2) - \hat{m}_f(n_1, n_2)) \quad (2.28)$$

où

$$\hat{m}_f(n_1, n_2) = \hat{m}_g(n_1, n_2) = \frac{1}{N} \sum_1^N g(n_1, n_2) \quad (2.29)$$

$$a(n_1, n_2) = \begin{cases} \frac{\hat{\sigma}_g^2(n_1, n_2) - \sigma_n^2}{\hat{\sigma}_g^2(n_1, n_2)} & \text{pour } \hat{\sigma}_g^2(n_1, n_2) > \sigma_n^2 \\ 0 & \text{ailleurs} \end{cases} \quad (2.30)$$

avec

$$\hat{\sigma}_g^2(n_1, n_2) = \frac{1}{N} \sum_1^N (g(n_1, n_2) - \hat{m}_g(n_1, n_2))^2 \quad (2.31)$$

où N est nombre des pixels, qui ont les mêmes caractéristiques que le pixel central au traitement.

Cette approche de segmentation donne une très bonne performance en comparaison avec les autres algorithmes existants. En fait, nous pouvons la considérer

comme une combinaison entre l'algorithme utilisant les caractéristiques locales [13] et l'approche utilisant le filtre Sigma [14]. Elle est meilleure que l'algorithme utilisant les statistiques locales grâce à la segmentation. Elle est supérieure à l'approche du filtre Sigma parce que, dans la façon de calculer les pixels traités, elle se base sur la puissance du bruit et celle de l'image, tandis que le filtre Sigma se base sur le principe du filtre moyen en supposant que l'image locale soit constante. Cette technique de segmentation est capable de s'adapter aux contours et lignes non seulement dans les directions principales mais aussi sur toute autre direction. Elle réduit le bruit autour des contours aussi bien que dans les zones plates parce que la segmentation s'adapte à tous les angles possibles. Alors avec cet algorithme on peut travailler avec toutes sortes d'images. La nature et la structure de l'image n'influencent pas beaucoup sa performance. (Les résultats sont présentés dans la section 2.3.).

2.2.2 Algorithmes dérivés de l'approche de segmentation

En comparaison avec les trois autres algorithmes présentés dans la section 2.1, la technique utilisant la segmentation permet une performance supérieure pour des images de natures diverses. Cependant quelquefois, dans certains cas où l'image contient beaucoup de zones plates et la variance du bruit est élevée, l'algorithme de segmentation aussi bien que l'approche utilisant les caractéristiques locales de Lee laisseront une certaine quantité de bruit dans de telles régions. Cet effet s'explique par le fait que si la variance du bruit est élevée, sa moyenne à l'intérieur d'une fenêtre de traitement n'est pas vraiment nulle, même si le bruit est un bruit gaussien de moyenne nulle. Dans l'algorithme de segmentation, il y a deux facteurs qui influencent beaucoup la performance. Le premier est la dimension de la fenêtre de traitement et le deuxième est la valeur de seuil précise S utilisée pour la segmentation. Normalement, ces facteurs sont fixés à l'avance dans le système de segmentation.

2.2.2.1 Approche utilisant la moyenne des images

L'idée principale du réducteur proposé dans cette section et les prochaines sections est fondée sur l'algorithme de segmentation. Selon l'algorithme de segmentation, la dimension de la fenêtre de traitement et la valeur de seuil S utilisée pour faire la segmentation sont fixes, tandis que les structures sur les images et les niveaux de bruit ajoutés changent d'une image à une autre. Alors, afin de s'adapter mieux aux caractéristiques de l'image contaminée et du bruit, nous utilisons cette technique sur plusieurs fenêtres avec des valeurs de seuils différentes. Cette approche est illustrée à la figure 2.10.

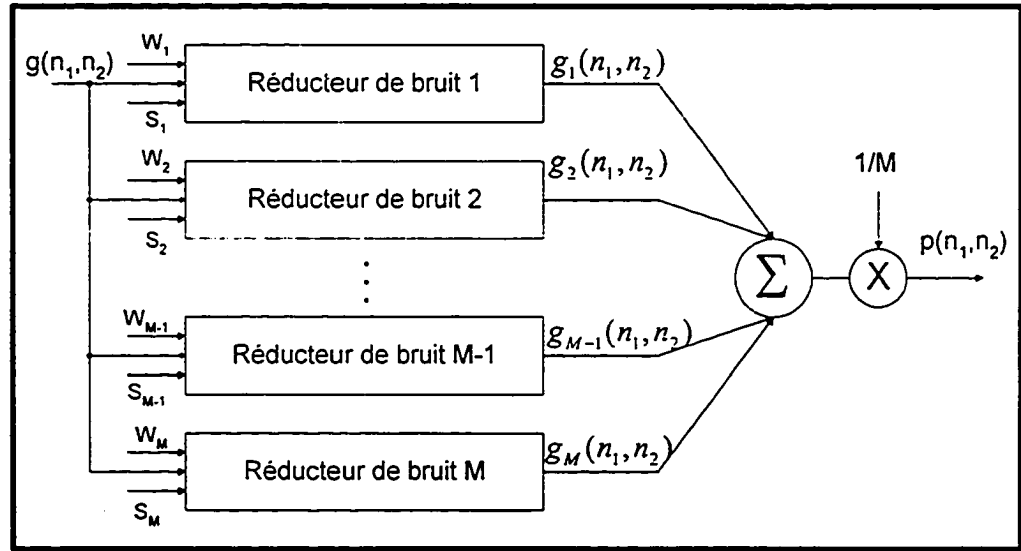


Figure 2.10 : Algorithme utilisant la moyenne des images

Nous pouvons considérer cet algorithme comme un système parallèle de plusieurs réducteurs de bruit utilisant la segmentation. Chaque réducteur a sa propre dimension de fenêtre de traitement, sa propre valeur de seuil et produit sa propre image de sortie $g_i(n_1, n_2)$. En prenant la moyenne de ces images de sortie, on obtient l'image traitée $p(n_1, n_2)$, qui permet une meilleure performance que n'importe quelle image de sortie $g_i(n_1, n_2)$. En effet, on obtient les équations suivantes :

$$p(n_1, n_2) = \frac{1}{M} \sum_{i=1}^M g_i(n_1, n_2) \quad (2.32)$$

où

$$g_i(n_1, n_2) = \hat{m}_{f_i}(n_1, n_2) + a_i(n_1, n_2)(g(n_1, n_2) - \hat{m}_{f_i}(n_1, n_2)) \quad (2.33)$$

avec

$$\hat{m}_{f_i}(n_1, n_2) = \hat{m}_{g_i}(n_1, n_2) = \frac{1}{N_i} \sum_{j=1}^{N_i} g(n_1, n_2) \quad (2.34)$$

$$a_i(n_1, n_2) = \begin{cases} \frac{\hat{\sigma}_{g_i}^2(n_1, n_2) - \sigma_n^2}{\hat{\sigma}_{g_i}^2(n_1, n_2)} & \text{pour } \hat{\sigma}_{g_i}^2(n_1, n_2) > \sigma_n^2 \\ 0 & \text{ailleurs} \end{cases} \quad (2.35)$$

$$\hat{\sigma}_{g_i}^2(n_1, n_2) = \frac{1}{N_i} \sum_{j=1}^{N_i} (g(n_1, n_2) - \hat{m}_{g_i}(n_1, n_2))^2 \quad (2.36)$$

Dans le sens du rapport signal sur bruit, cette approche présente un meilleur résultat que n'importe quelle technique présentée précédemment. Même au niveau de la qualité visuelle (section 2.3), l'image traitée par cet algorithme est meilleure. Cependant, quand le niveau de bruit ajouté devient excessivement élevé, il reste toujours une certaine quantité du bruit dans les zones plates de l'image. L'approche présentée dans la prochaine section a pour but d'améliorer encore la qualité visuelle dans de telles régions.

2.2.2.2 Approche utilisant une image constante

La technique proposée dans cette section utilise aussi le système parallèle de plusieurs réducteurs appliquant la segmentation à la différence d'utiliser non seulement la moyenne des images de sortie $g_i(n_1, n_2)$ mais également la moyenne des images de la

moyenne $\hat{m}_{g_i}(n_1, n_2)$ dans ses calculs. La figure 2.11 ci-dessous représente la structure de ce système.

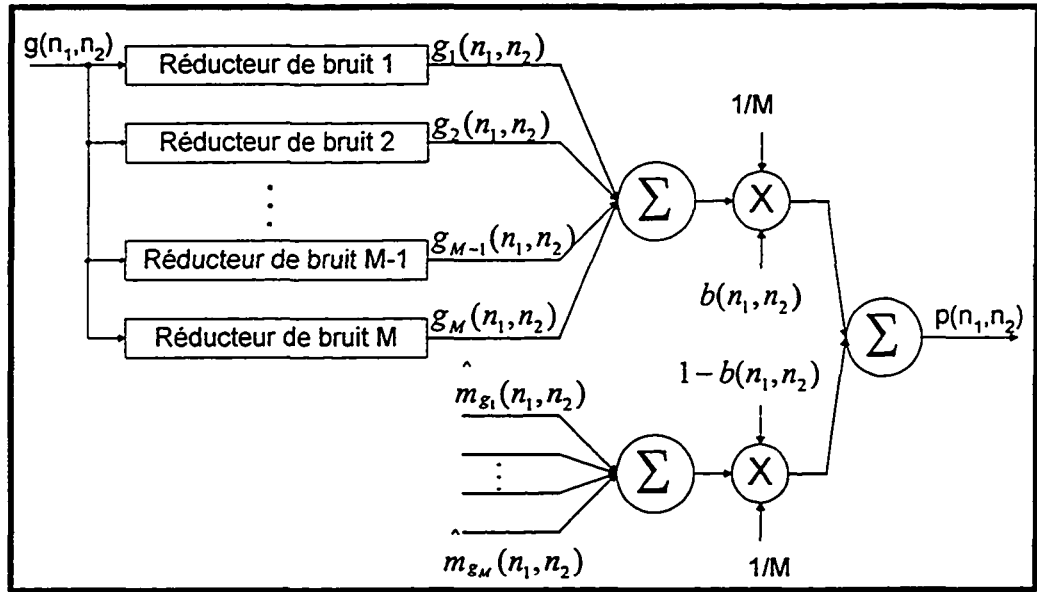


Figure 2.11 : Algorithme utilisant une image constante

Dans ce système, la façon de calculer l'image traitée $p(n_1, n_2)$ est fondée sur les équations suivantes :

$$p(n_1, n_2) = \frac{1}{M} b(n_1, n_2) \sum_{i=1}^M g_i(n_1, n_2) + \frac{1}{M} [1 - b(n_1, n_2)] \sum_{i=1}^M \hat{m}_{g_i}(n_1, n_2) \quad (2.37)$$

où

$$b(n_1, n_2) = \frac{\frac{1}{M} \sum_{i=1}^M \sigma_{f_i}^2(n_1, n_2)}{\frac{1}{M} \sum_{i=1}^M \sigma_{g_i}^2(n_1, n_2)} = \frac{\frac{1}{M} \sum_{i=1}^M \sigma_{g_i}^2(n_1, n_2) - \sigma_n^2}{\frac{1}{M} \sum_{i=1}^M \sigma_{g_i}^2(n_1, n_2)} \quad (2.38)$$

$$\hat{m}_{f_i}(n_1, n_2) = \hat{m}_{g_i}(n_1, n_2) = \frac{1}{N_i} \sum_{j=1}^{N_i} g(n_1, n_2) \quad (2.39)$$

$$\hat{\sigma}_{g_i}^2(n_1, n_2) = \frac{1}{N_i} \sum_{j=1}^{N_i} (g(n_1, n_2) - \hat{m}_{g_i}(n_1, n_2))^2 \quad (2.40)$$

En ce qui concerne de la qualité visuelle dans les zones plates, cette approche permet une meilleure performance grâce à l'utilisation des images de la moyenne $\hat{m}_g(n_1, n_2)$. Cependant, dans le sens du rapport signal sur bruit, les résultats sont variables dépendamment de la structure des images à traiter. Pour les images contenant beaucoup de zones plates, cet algorithme domine la technique « moyenne ». Par contre, quand les images dégradées contiennent beaucoup de zones de contours et de zones de détails fins, son rapport signal sur bruit est plus petit que celui de l'approche « moyenne ». Alors, une meilleure approche consiste à combiner ces deux algorithmes.

2.2.2.3 Approche combinée

L'idée de cette technique combinée est simple. Dans les zones plates de l'image, on utilise les pixels de l'approche « constante ». Par contre, les pixels de la technique « moyenne » sont appliqués pour les zones contenant beaucoup de contours et de détails fins. Afin de distinguer les zones plates et les zones de contours et de détails fins, il faut faire une détection de composantes à haute fréquence. Normalement, la détection de contours et de détails fins est une tâche difficile en présence du bruit. Dans cette approche, nous n'appliquons pas la détection sur l'image dégradée mais sur l'image de sortie traitée par la technique « constante ». Cette image est très lisse dans les zones plates et grâce à cela elle permet un bon résultat dans la détection de composantes à haute fréquence. Mathématiquement, ce système « combiné » peut être représenté par l'équation 2.41 :

$$p(n_1, n_2) = e(n_1, n_2)p_{OM}(n_1, n_2) + (1 - e(n_1, n_2))p_{OC}(n_1, n_2) \quad (2.41)$$

où $p_{OM}(n_1, n_2)$ et $p_{OC}(n_1, n_2)$ sont, respectivement, les images de sortie traitées par la technique « moyenne » et la technique « constante ». Dans cette équation, $e(n_1, n_2)$ représente l'image à haute fréquence créée par la détection de contours et de détails fins. Dans la littérature [5,26], il existe une grande variété des techniques de détection

de contours. Nous utilisons, dans ce projet, les opérateurs horizontal et vertical de Sobel et un filtre passe-bas qui sont des filtres bidimensionnels à réponse impulsionnelle (FIR) définis par les équations qui suivent :

$$SB_H = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad SB_V = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \quad (2.42)$$

$$PB = \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} / 16 \quad (2.43)$$

Les filtres SB_H et SB_V correspondent respectivement à la dérivée horizontale et verticale. Ils servent à la détection des transitions horizontale et verticale. Leurs réponses sont bipolaires. Nous nous intéressons seulement à l'amplitude de ces réponses. Ainsi, une fonction de valeur absolue est appliquée à la sortie de chacun de ces opérateurs. La réponse globale est la somme de ces valeurs absolues. Ensuite, cette réponse est filtrée par le filtre passe-bas PB . Étant donné que les défauts dans les zones plates sont en général de faible amplitude, la discrimination entre les vrais contours et les détails se fait par une limitation de seuil T . Si $d(n_1, n_2)$ est la détection de contours au pixel de coordonnée (n_1, n_2) nous pouvons écrire :

$$d(n_1, n_2) = \begin{cases} 1 & \text{pour } r(n_1, n_2) > T \\ 0 & \text{ailleurs} \end{cases} \quad (2.44)$$

où $r(n_1, n_2)$ représente la réponse après le filtre passe-bas PB .

La figure 2.12 ci-dessous illustre le schéma-bloc du système combiné utilisant la détection de contours.

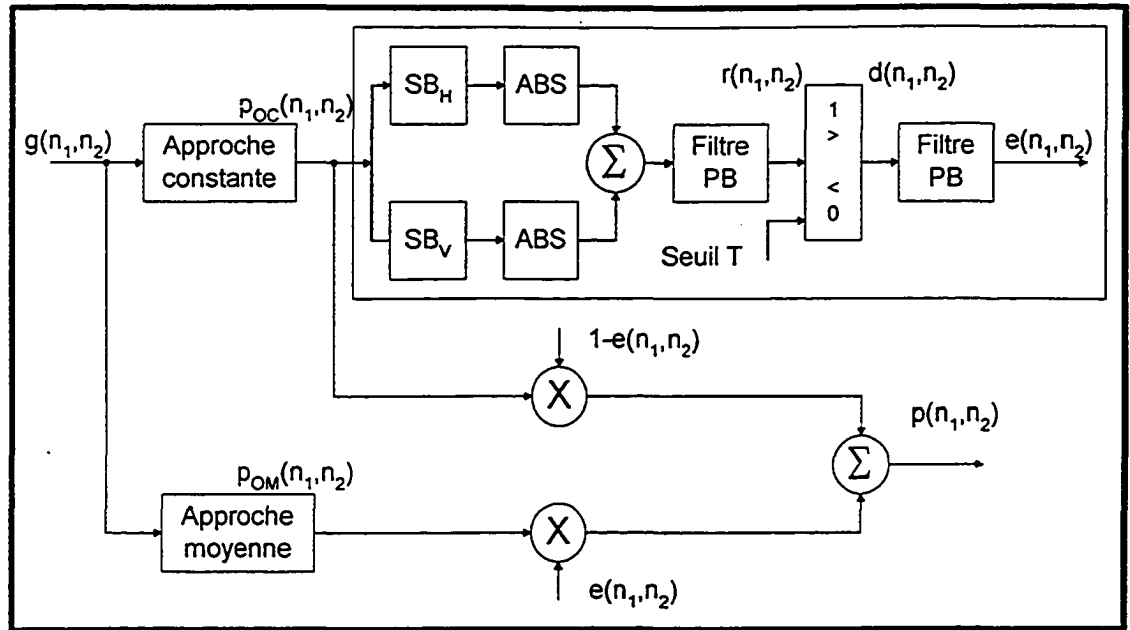


Figure 2.12 : Algorithme combiné

En prenant les avantages de la technique « moyenne » aussi bien que ceux de l'approche « constante », cet algorithme permet une très belle performance en comparaison avec les autres techniques. Il est robuste pour toutes les images et garantit un résultat supérieur dans deux sens : la qualité visuelle et le rapport signal sur bruit. La prochaine section est réservée à la comparaison de la performance entre les réducteurs existants et ceux proposés.

2.3 Comparaison de la performance entre les réducteurs de bruit

Dans les dernières sections, nous avons présenté le principe et la structure de plusieurs systèmes de réduction de bruit aléatoire additif. Dans cette section, nous présentons une comparaison détaillée de la performance entre eux. En effet, la performance de chaque algorithme se modifie en fonction de plusieurs facteurs. C'est pourquoi dans la pratique, la comparaison de toutes les techniques de réduction de bruit est une tâche difficile. Normalement, chaque algorithme peut s'avérer efficace pour une seule sorte d'images et il dépend beaucoup du niveau de bruit ajouté.

Dans le domaine de traitement des images, on définit deux principaux critères pour juger la performance de chaque système de réduction de bruit. L'un est qualitatif et l'autre est quantitatif. Le critère qualitatif a pour objet de juger la qualité visuelle de l'image. Selon ce critère, la qualité visuelle est normalement évaluée par deux facteurs suivantes: l'efficacité dans la réduction de bruit et la conservation des contours et des détails fins. Pour quantifier la performance des systèmes, certaines mesures objectives peuvent être définies, comme l'erreur quadratique moyenne et le rapport signal sur bruit. Normalement, si l'image originale est disponible, l'erreur quadratique moyenne entre l'image originale et l'image dégradée aussi bien que celle entre l'image originale et l'image traitée sont calculées par les équation suivantes:

$$MSE_{dégradé} = Var[f(n_1, n_2) - g(n_1, n_2)] \quad (2.45)$$

$$MSE_{traité} = Var[f(n_1, n_2) - p(n_1, n_2)] \quad (2.46)$$

Et les rapports signal sur bruit de l'image dégradée et de l'image traitée sont obtenus par les formules suivantes :

$$PSNR_{dégradé} = 10 \log_{10} \frac{255^2}{MSE_{dégradé}} \quad (dB) \quad (2.47)$$

$$PSNR_{traité} = 10 \log_{10} \frac{255^2}{MSE_{traité}} \quad (dB) \quad (2.48)$$

L'amélioration du rapport signal sur bruit est calculée par l'équation 2.49 :

$$\Delta PSNR = PSNR_{traité} - PSNR_{dégradé} \quad (dB) \quad (2.49)$$

Dans le domaine de la restauration d'images, le rapport signal sur bruit est une mesure objective importante pour juger de la qualité de l'estimation obtenue. Plus ce rapport est grand, plus l'image traitée est proche de celle de l'image originale. En général, ce rapport signal sur bruit influence aussi la qualité visuelle de l'image. Si une personne observe deux images dégradées par le même type de bruit, elle va juger que l'image avec un plus grand rapport signal sur bruit est plus proche à celle de l'image originale [16].

Comme nous l'avons mentionné dans le premier chapitre, la qualité de l'image traitée par un système de réduction de bruit dépend beaucoup de la structure naturelle de l'image et du niveau de bruit ajouté. Pour avoir une comparaison raisonnable et détaillée, nous avons testé les algorithmes présentés en haut avec plusieurs images qui sont très différentes par leur nature et viennent représenter plusieurs classes d'images. Ce sont les images standards utilisées souvent dans la littérature et dans les tests : « Airplane », « Lenna », « Couple » et « Bridge ». En même temps nous avons utilisé aussi plusieurs niveaux de bruit différents. La variance de ce bruit ajouté varie entre 36 et 400. L'évaluation de la performance des techniques est fondée sur les paramètres qui suivent :

- le rapport signal sur bruit de chaque image traitée;
- l'efficacité dans la réduction de bruit;
- la conservation des contours et des détails fins;
- la complexité de chaque système.

2.3.1 Le rapport signal sur bruit

Le tableau 2.1 et les figures 2.13, 2.14 et 2.15 présentent les rapports signal sur bruit des images dégradées ainsi que l'amélioration du *PSNR* des images traitées par tous les algorithmes présentés en haut, avec quatre images différentes. Selon ces figures, nous voyons que la technique des quatre filtres unidimensionnels de Chan et Lim aussi bien que celle utilisant le filtre Sigma, ne fonctionnent pas bien pour les images contenant beaucoup de contours et beaucoup de détails fins comme l'image « Couple » et l'image « Bridge ». L'amélioration du rapport signal sur bruit est même négative dans les cas des images « Couple » et « Bridge » quand le niveau de bruit est petit. Par contre, ces deux approches donnent de bons résultats pour les images contenant beaucoup des zones plates comme l'image « Airplane » et l'image « Lenna », et dégradées par un niveau de bruit plus important. Selon ces mesures objectives,

l'approche utilisant les statistiques locales est globalement satisfaisante pour toute sorte d'images et de niveaux de bruits ajoutés.

En regardant l'algorithme de segmentation, nous pouvons constater que cette technique est excellente, et supérieure aux trois premières. Elle est très robuste pour toutes les images et tous les niveaux de bruits. L'amélioration de *PSNR* est plus grande que celle des trois autres de 0.5 jusqu'à 2 dB ou plus. Les trois dernières approches « moyenne », « constante » et « combinée » sont les améliorations de la technique de segmentation. En fait, la différence entre elles n'est pas tellement considérable (Figure 2.15). Leurs résultats sont entre 0.2 et 2.4 dB plus grands que celui de l'algorithme de segmentation dépendant de l'image à traiter et du niveau de bruit ajouté.

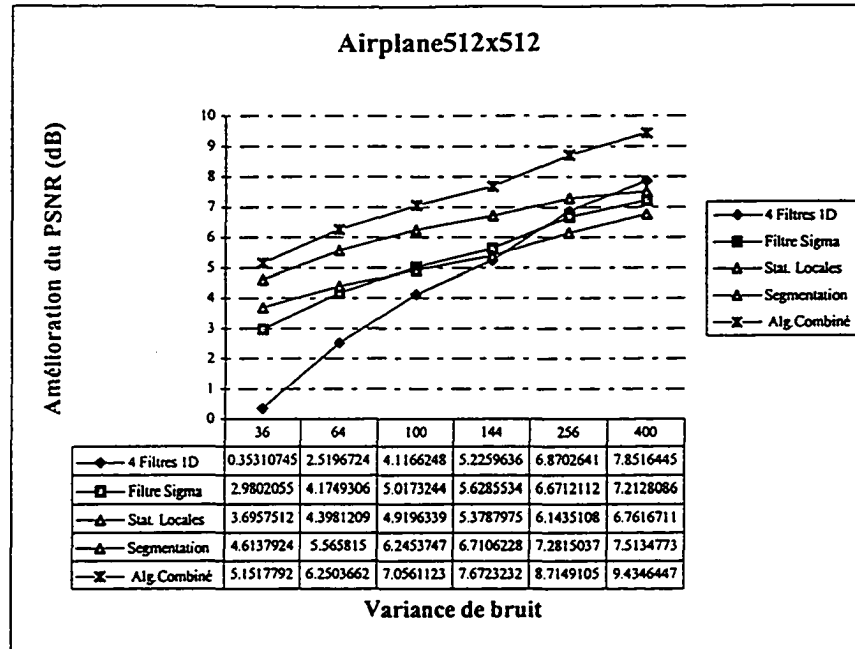
Tableau 2.1 : Rapports signal sur bruit des images dégradées en dB

	$\sigma^2=36$	$\sigma^2=64$	$\sigma^2=100$	$\sigma^2=144$	$\sigma^2=256$	$\sigma^2=400$
Airplane	32.538216	30.047840	28.118032	26.562525	24.086069	22.184742
Lenna	32.547421	30.068775	28.157534	26.619537	24.182871	22.279392
Couple	32.553028	30.066629	28.145214	26.595711	24.116877	22.186773
Bridge	32.560387	30.079086	28.157898	26.609459	24.145622	22.226210

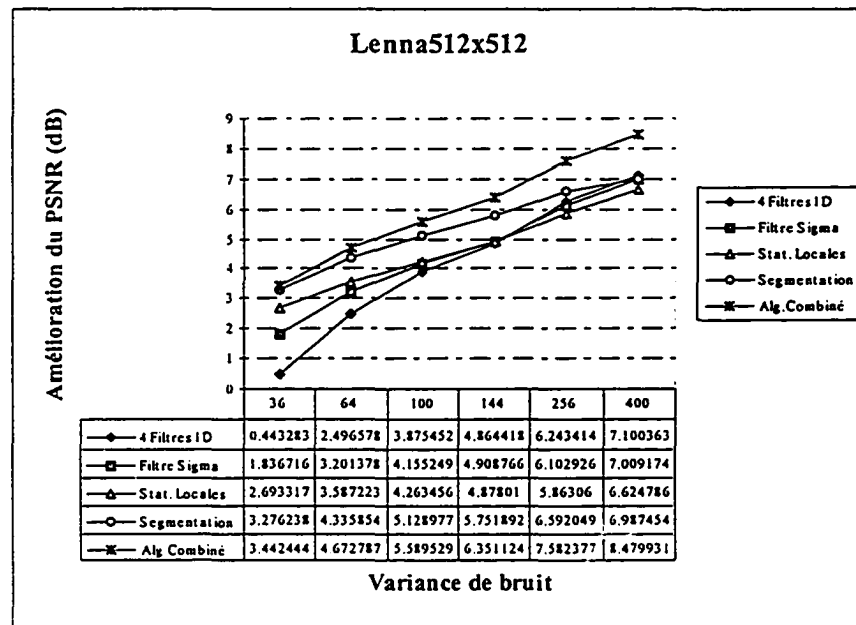
2.3.2 L'efficacité de la réduction de bruit

Ce critère subjectif a pour objet de juger si une image traitée est bien lissée. En général, on regarde les zones plates de l'image et on évalue s'il y a encore le bruit. Normalement, le bruit présent dans des zones plates est facilement détecté par l'œil humain. L'algorithme des quatre filtres unidirectionnels et l'approche utilisant le filtre Sigma fonctionnent très bien selon ce critère. Les zones plates des images traitées par ces techniques sont très glisses et deviennent agréables pour la qualité visuelle. En regardant les images traitées par l'approche utilisant les caractéristiques locales et l'algorithme de segmentation, on peut voir que dans les zones plates il reste encore quantité de bruit. Dans le cas où le niveau de bruit n'est pas grand, la qualité visuelle est toujours bonne, mais quand le niveau de bruit devient très important, cette qualité s'en

ressent. Ce problème est résolu par les trois dernières approches. Dans ce critère, la meilleure approche est celle utilisant une image constante présentée à la section 2.2.2.2.



a)

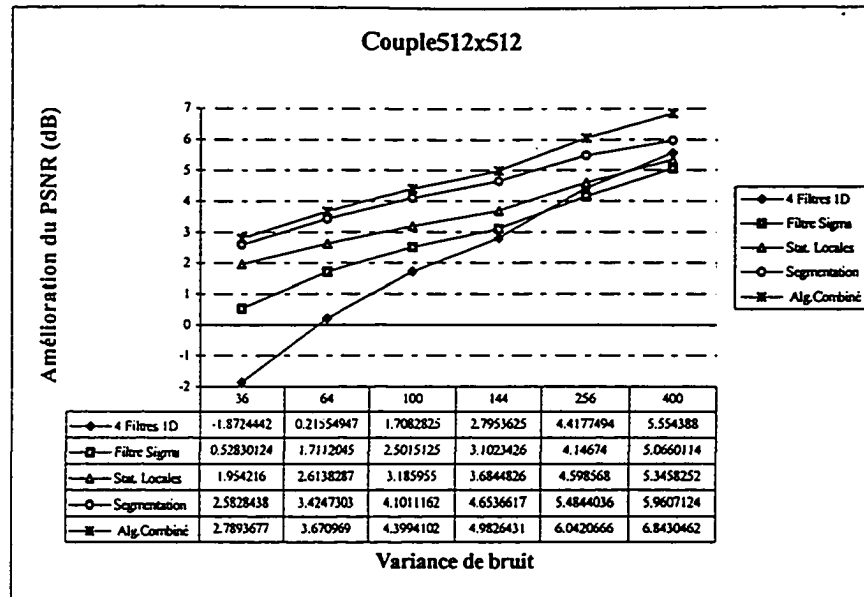


b)

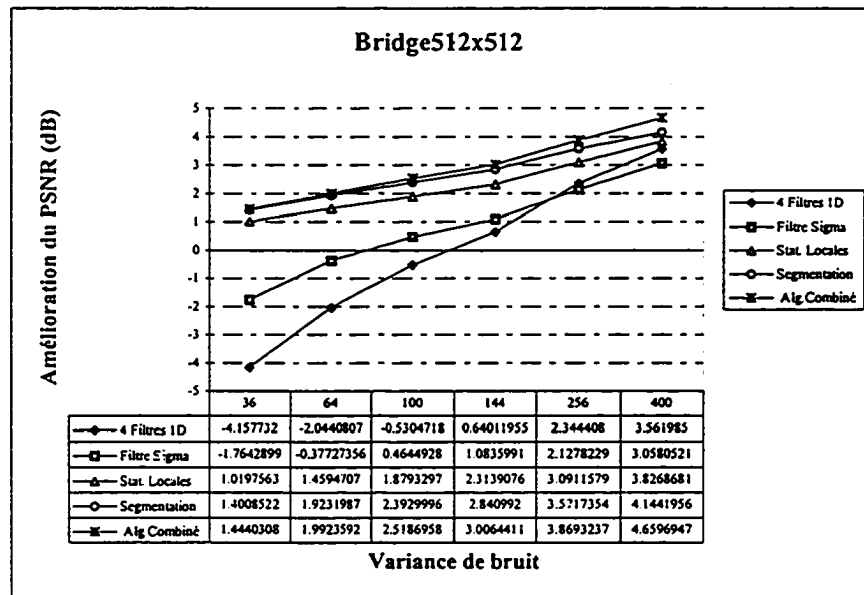
Figure 2.13 : Comparaison des rapports signal sur bruit

a) Image «Airplane » avec dimension de 512x512

b) Image « Lenna » avec dimension de 512x512



c)

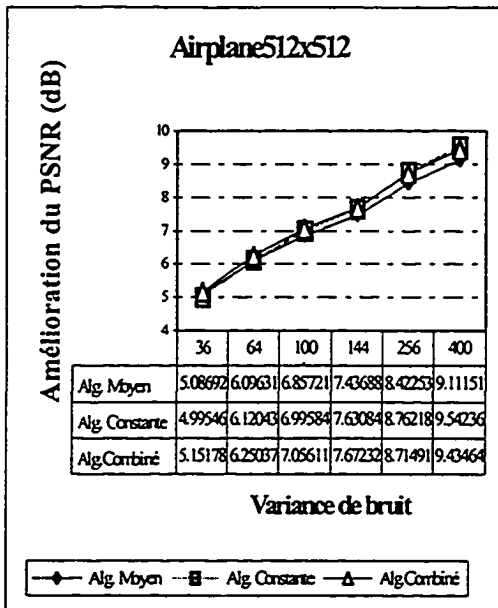


d)

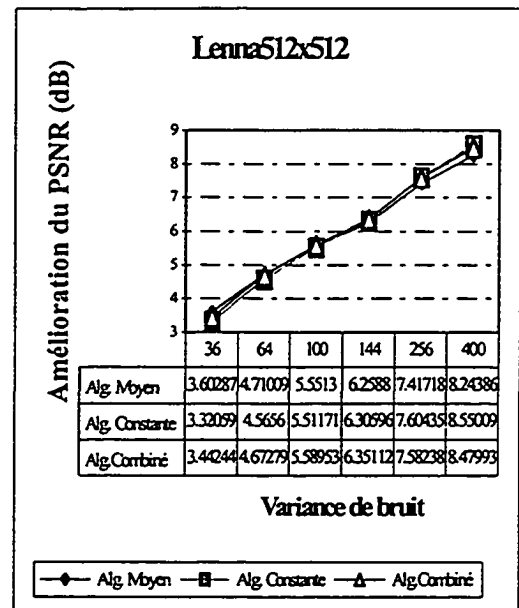
Figure 2.14 : Comparaison des rapports signal sur bruit

c) Image « Couple » avec dimension de 512x512

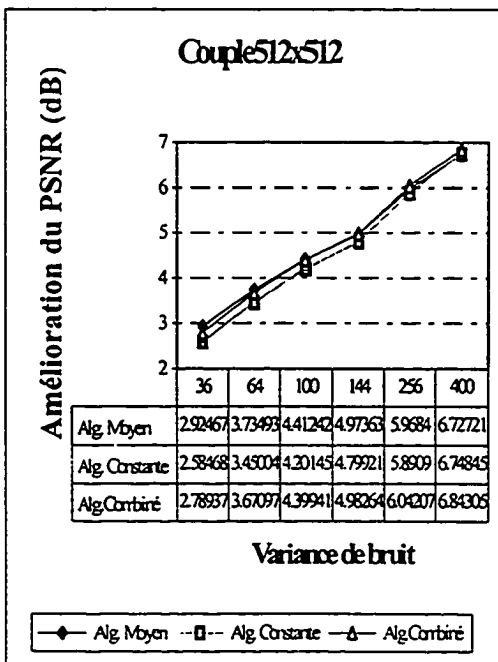
d) Image « Bridge » avec dimension de 512x512



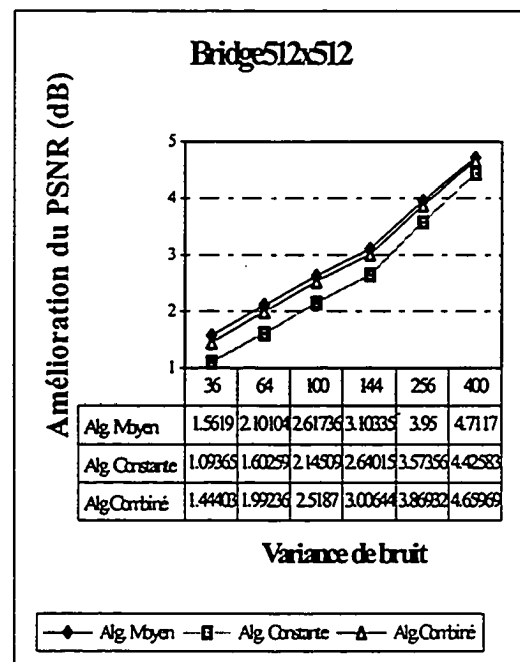
a)



b)



c)



d)

Figure 2.15 : Comparaison des rapports signal sur bruit entre les trois algorithmes utilisant la parallèle de l'approche de segmentation.

a) Image « Airplane »

b) Image « Lenna »

c) Image « Couple »

d) Image « Bridge »

2.3.3 La conservation des contours et des détails fins

Pour la perception visuelle humaine, la présence des contours et des lignes joue un rôle important. Tous les algorithmes présentés ont pour objet de préserver les contours mais les résultats entre eux sont très différents. L'approche de Chan et Lim préserve seulement des lignes et des contours qui se trouvent dans les quatre directions principales. On perd les contours qui sont dans les autres directions. De plus, à cause du traitement directionnel, cet algorithme produit même des faux contours. La technique utilisant le filtre Sigma est efficace dans la conservation des contours et des détails fins seulement quand la différence d'intensité entre eux et les textures est plus grande que 2σ . Sinon, tous les contours et les détails fins sont enlevés de l'image. Nous pouvons considérer que les images traitées par cet algorithme contiennent des brouillages et ne sont pas nettes comme celles des autres. L'algorithme utilisant les statistiques locales garde des contours et des détails fins mais ne réduit pas le bruit à proximité. Ce bruit dégrade des contours et réduit la qualité visuelle de l'image.

Dans le sens de la conservation des contours et des détails fins, nous voyons que l'approche de segmentation et les trois dernières approches sont plus efficaces que les autres techniques. Elles gardent très bien les contours et ne produisent aucun effet secondaire. Ces résultats sont obtenus grâce à la segmentation locale qui permet une bonne adaptation aux caractéristiques des images et du bruit.

2.3.4 La complexité

Tous les algorithmes présentés sont non-récurifs mais leurs complexités sont différentes. L'algorithme utilisant le filtre Sigma est le plus rapide parce qu'il ne demande que le calcul de la moyenne de 2σ et de la moyenne des pixels voisins. L'approche qui nécessite le plus de calcul est la technique combinée. Dans nos tests, la fenêtre de traitement pour l'algorithme utilisant les statistiques locales, la technique des quatre filtres unidimensionnels, l'approche utilisant le filtre Sigma et l'algorithme de

segmentation a une dimension de 5x5. Les algorithmes utilisant l'approche parallèle à la segmentation ont les fenêtres de dimension 5x5 et 7x7 avec deux valeurs de seuil de σ et $\sigma+2$ où σ est l'écart-type du bruit ajouté. La complexité des algorithmes est représentée par le tableau 2.2 :

Tableau 2.2 : Complexité des algorithmes

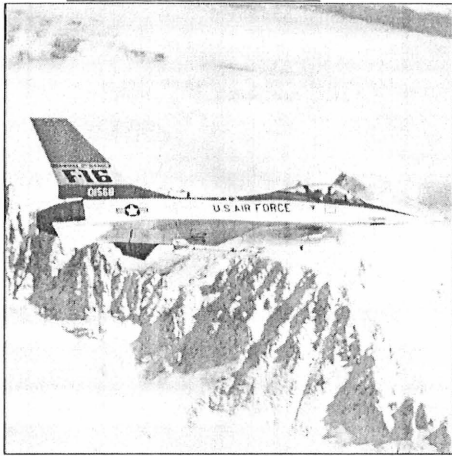
	Nombre des opérations pour chacun pixel traité	Rapport de complexité
Filtre Sigma	62	1
Stat. Locales	83	1.34
Segmentation	133	2.15
4 Filtres 1D	157	2.53
Alg. Moyen	562	9.01
Alg. Constant	584	9.42
Alg. Combiné	643	10.41

2.3.5 Sommaire

En conclusion, si on ne tient pas de la complexité, le réducteur de bruit qui donne une meilleure performance est l'approche combinée. Il garantit non seulement un plus grand rapport signal sur bruit, mais aussi une bonne qualité visuelle. Les quatre images traitées par cet algorithme sont plus lisses, plus nettes et plus proches des images originales que les autres.

Il est bon de mentionner que certains résultats de ce travail constituent l'objet d'une demande de brevet [12].

Dans les prochaines pages, nous présentons visuellement les images de résultats avec leurs agrandissements. Les images contaminées sont dégradées par bruit gaussien de moyenne et d'une variance de 400.



a)



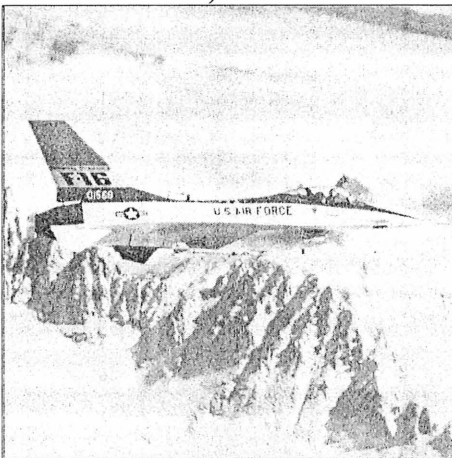
b)



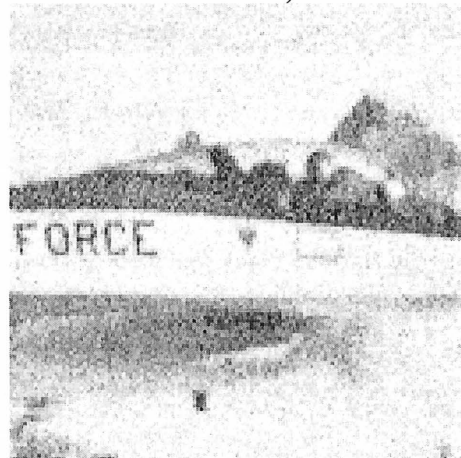
c)



d)



e)



f)

Figure 2.16 : Image « Airplane » dégradée par le bruit gaussien additif.

a) Image sans bruit

b) Agrandissement

c) Avec bruit de variance de 100

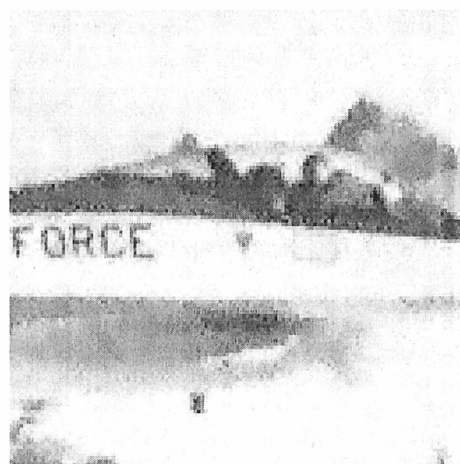
d) Agrandissement

e) Avec bruit de variance de 400

f) Agrandissement



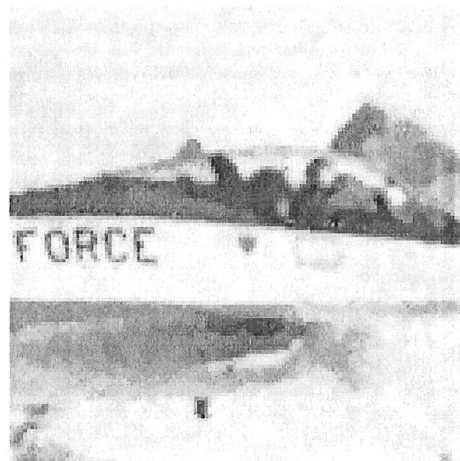
a)



b)



c)



d)



e)



f)

Figure 2.17 : Résultats de l'image « Airplane » ($\sigma^2=400$) traitée par les algorithmes.

a) Alg. de quatre filtres 1-D

b) Alg. des statistiques locales

c) Alg. du filtre Sigma

d) Alg. de segmentation

e) Approche moyenne

f) Approche combinée



a)



b)



c)



d)



e)



f)

Figure 2.18 : Image « Lenna » dégradée par le bruit gaussien additif.

a) Image sans bruit

b) Agrandissement

c) Avec bruit de variance de 100

d) Agrandissement

e) Avec bruit de variance de 400

f) Agrandissement



a)



b)



c)



d)



e)



f)

Figure 2.19 : Résultats de l'image « Lenna » ($\sigma^2=400$) traitée par les algorithmes.

a) Alg. de quatre filtres 1-D

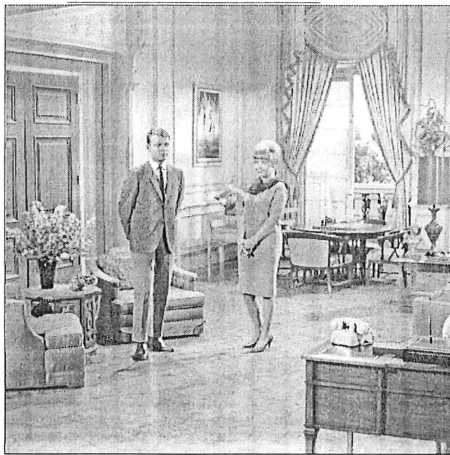
b) Alg. des statistiques locales

c) Alg. du filtre Sigma

d) Alg. de segmentation

e) Approche moyenne

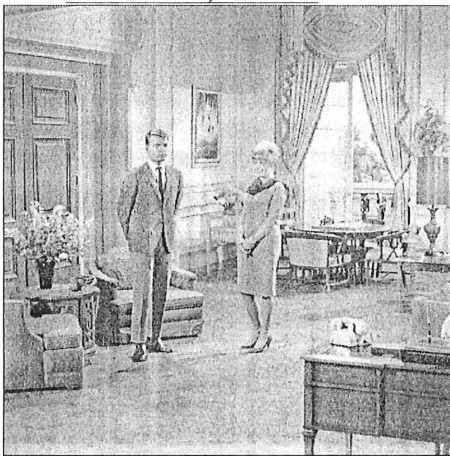
f) Approche combinée



a)



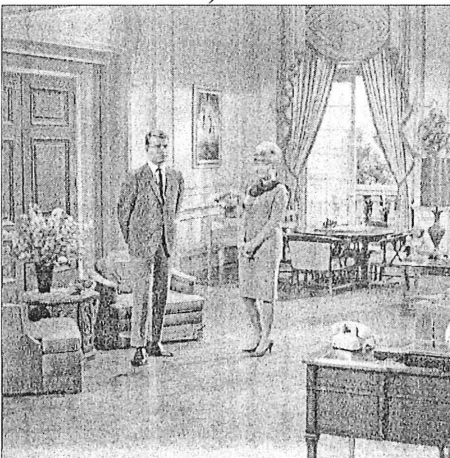
b)



c)



d)



e)



f)

Figure 2.20 : Image « Couple » dégradée par le bruit gaussien additif.

a) Image sans bruit

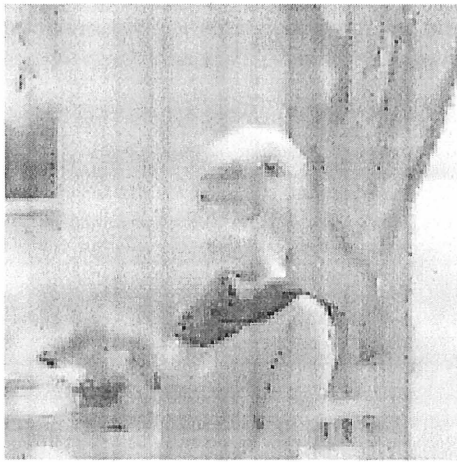
b) Agrandissement

c) Avec bruit de variance de 100

d) Agrandissement

e) Avec bruit de variance de 400

f) Agrandissement



a)



b)



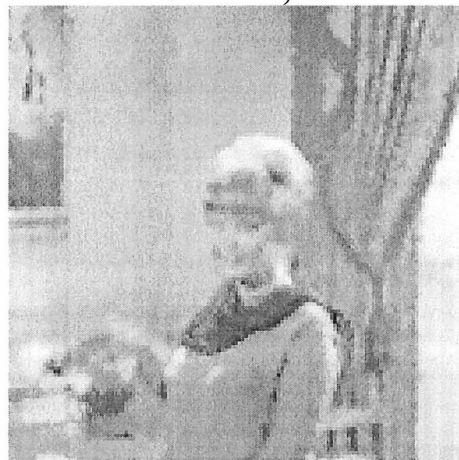
b)



c)



e)



f)

Figure 2.21 : Résultats de l'image « Couple » ($\sigma_n^2=400$) traitée par les algorithmes.

a) Alg. de quatre filtres 1-D

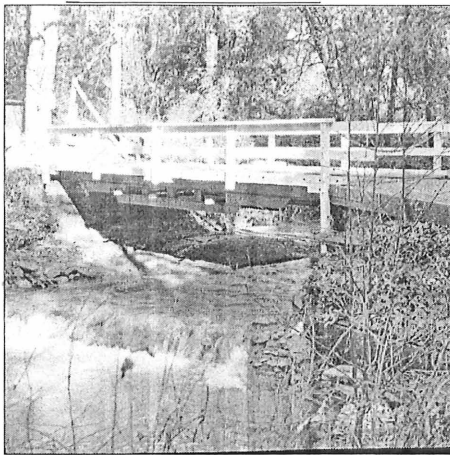
b) Alg. des statistiques locales

c) Alg. du filtre Sigma

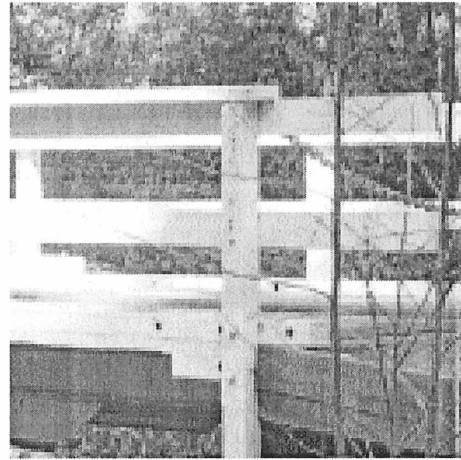
d) Alg. de segmentation

e) Approche moyenne

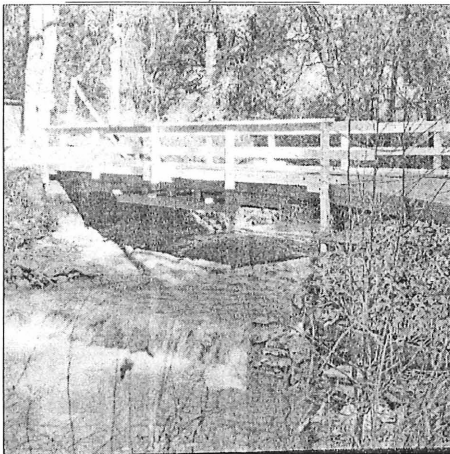
f) Approche combinée



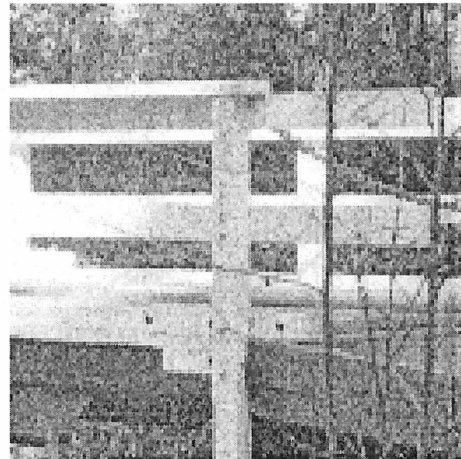
a)



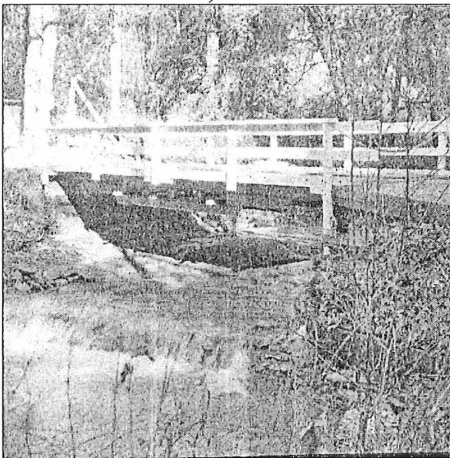
b)



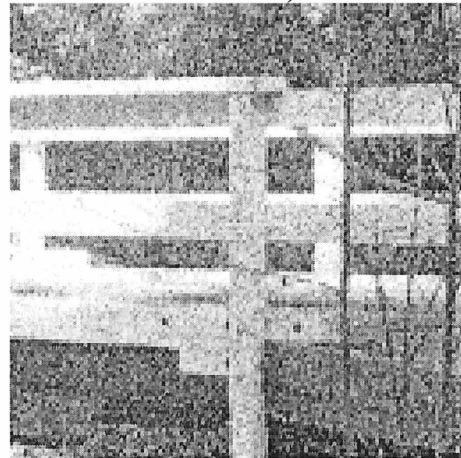
c)



d)



e)



f)

Figure 2.22 : Image « Bridge » dégradée par le bruit gaussien additif.

a) Image sans bruit

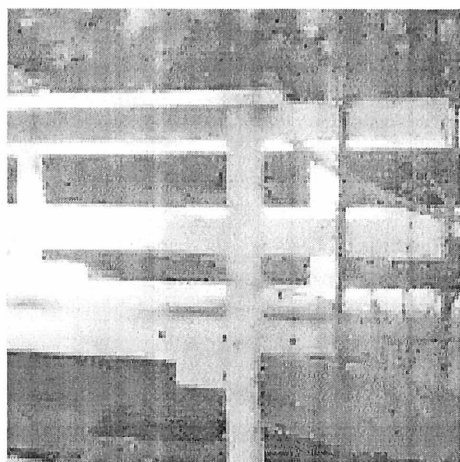
b) Agrandissement

c) Avec bruit de variance de 100

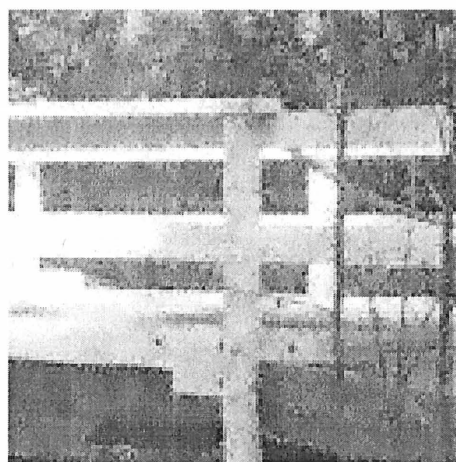
d) Agrandissement

e) Avec bruit de variance de 400

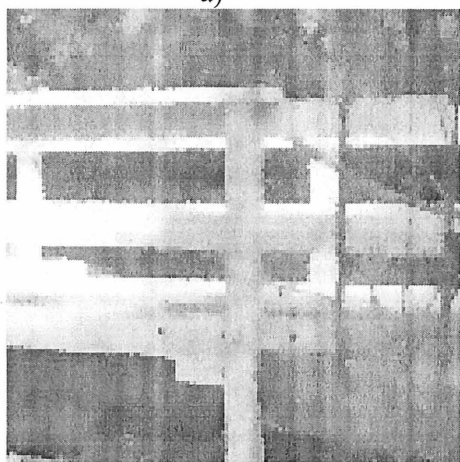
f) Agrandissement



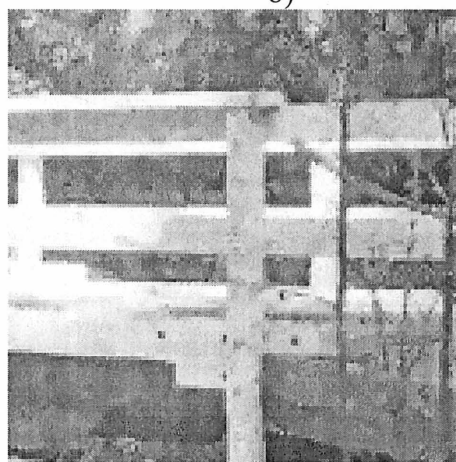
a)



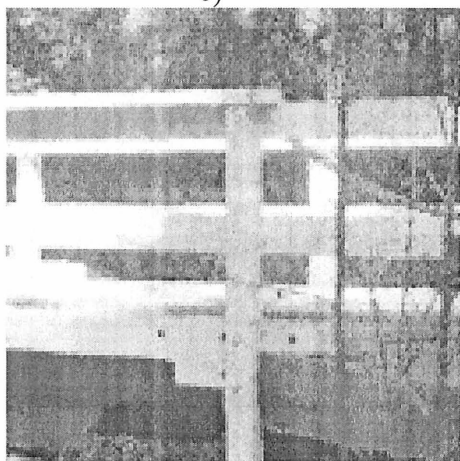
b)



c)



d)



e)



f)

Figure 2.23 : Résultats de l'image « Bridge » ($\sigma_n^2=400$) traitée par les algorithmes.

a) Alg. de quatre filtres 1-D

b) Alg. des statistiques locales

c) Alg. du filtre Sigma

d) Alg. de segmentation

e) Approche moyenne

f) Approche combinée

Chapitre 3

Post-traitement pour codec MPEG-2

Ce chapitre se compose de trois sections. Afin d'obtenir une connaissance du codec MPEG-2 et de faciliter la compréhension des défauts causés par ce standard, nous commençons par présenter les notions de base de cet algorithme, le fonctionnement de l'encodeur et du décodeur, ainsi que les techniques de compression employées. Par la suite, nous effectuons une étude sur les défauts causés par le codec MPEG-2. Cette étude sert à identifier les effets de distorsion et à expliquer les sources qui sont responsables de ces erreurs. Enfin, la troisième et aussi dernière section du chapitre est réservée au post-traitement. Dans cette section, grâce aux réducteurs de bruit proposés au chapitre précédent, nous développons un système de post-traitement qui fonctionne à un débit de 3 à 8 Mbit/s. Ce post-traitement a pour but d'améliorer la qualité visuelle des images vidéo compressées par le standard MPEG-2.

3.1 Codec MPEG-2

Le standard MPEG-2 a pour objectif de définir une technique de compression pour le signal vidéo avec une haute qualité. Il peut opérer à divers débits et ce, en fonction de la qualité de l'image à la sortie. Une séquence vidéo avec un débit de 5 Mbit/s a une résolution de qualité NTSC/PAL, tandis qu'à la résolution utilisée dans les studios, la séquence doit être compressée avec un débit de 10 Mbit/s et plus.

À la différence du standard MPEG-1, la technique MPEG-2 supporte les deux types d'images : progressif et entrelacé. L'algorithme de base fonctionne normalement à la résolution d'environ de 720 pixels horizontalement et 480 pixels verticalement, et à la fréquence de 30 images par seconde. Dans la vidéo entrelacée, chaque image se compose de deux champs, l'un contient les lignes paires et l'autre contient les lignes impaires. À l'intérieur d'une image vidéo entrelacée, les lignes sont balayées d'une façon intercalée, c'est-à-dire : dans la dimension spatiale, elles sont adjacentes, mais dans la dimension temporelle elles ne le sont pas.

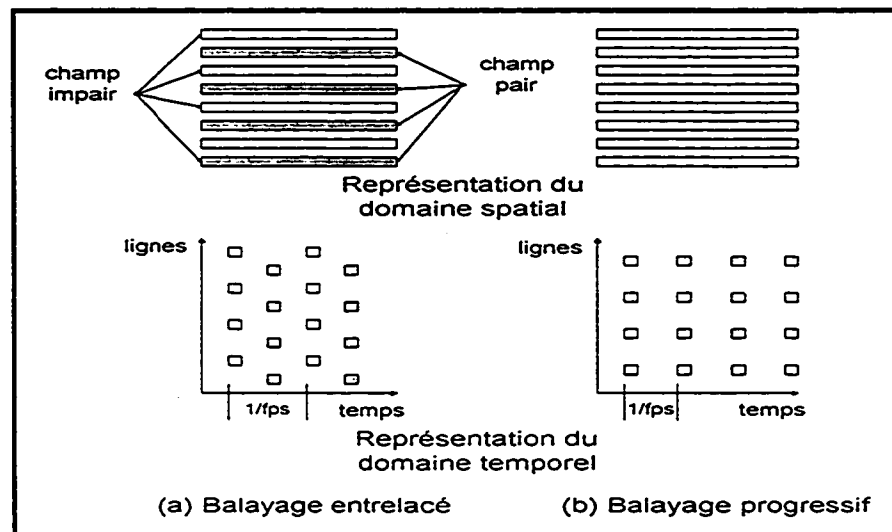


Figure 3.1 : Signal vidéo entrelacé et progressif

Dans les images vidéo progressives, par contre, il n'y a pas de notion de champ.

Les lignes sont adjacentes dans la dimension spatiale aussi bien que dans la dimension temporelle. Une image commence au coin gauche en haut et finit au coin droit en bas. La figure 3.1 illustre ces deux types d'images vidéo. Les champs dans l'image vidéo entrelacée peuvent être codés séparément comme des images de champ ou ils peuvent être interpolés et codés comme une seule image.

3.1.1 Types d'images

L'accès aléatoire et le taux élevé de compression sont les deux contraintes principales de la phase du développement des techniques MPEG. D'une part, l'accès aléatoire peut être réalisé en utilisant la technique du codage *intra-image*. D'autre part, le taux de compression peut être augmenté grâce à l'application de la technique du codage prédictif. Afin de satisfaire ces exigences, MPEG-2 définit trois types d'images qui sont similaires à ceux du standard MPEG-1 [15].

Comme dans le codec MPEG-1, toutes les images d'entrée du codec MPEG-2 peuvent être codées comme I-images, P-images ou B-images. Les I-images ou *images intra-codées* sont compressées en utilisant la technique du codage *intra-image*. C'est-à-dire : elles sont encodées indépendamment des autres images voisines. Cette technique garantit le mode « accès aléatoire » mais elle n'offre qu'un taux de compression moyen.

Les P-images ou *images codées prédictives* sont encodées par la technique de compensation de mouvement en utilisant l'information des I-images ou P-images précédentes. Le taux de compression des P-images est meilleur que les I-images et les P-images peuvent être utilisées pour la compensation de mouvement supplémentaire.

Les B-images ou *images codées prédictives bidirectionnelles* sont compressées de la même façon que celle des P-images, sauf qu'elles sont référencées par rapport P-images ou I-images précédentes et futures. Ces images ne sont pas utilisées pour la

prédiction de mouvement supplémentaire et elles permettent d'obtenir le taux de compression le plus élevé en comparaison avec les I-images ou P-images.

La figure 3.2 présente une séquence typique avec huit images et les relations entre elles. Les flèches indiquent la dépendance de chacune des images dans le codage prédictif unidirectionnel ou bidirectionnel. Les explications sur ce point sont données dans les prochaines sections.

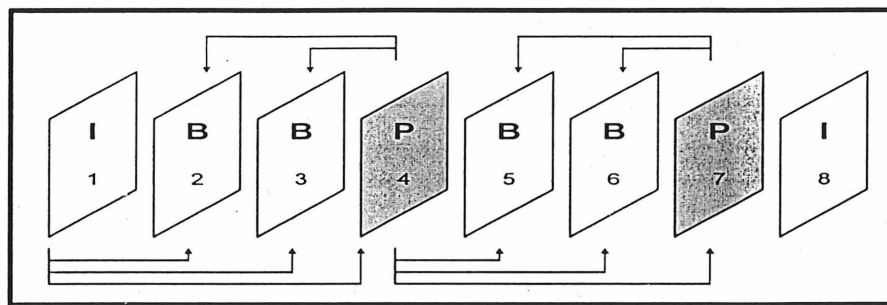


Figure 3.2 : I-, P-, et B-images dans une séquence de vidéo MPEG-2

3.1.2 Hiérarchie dans le système de codage

Comme le codage MPEG-1, le codage MPEG-2 peut être représenté hiérarchiquement par six couches, tel que montré par la figure 3.3.

La première est la couche de séquence. Elle se compose d'une entête de séquence et d'un ou de plusieurs groupes d'images. Dans l'entête de séquence, on trouve plusieurs informations nécessaires comme les dimensions horizontales et verticales de chaque image, la fréquence des images par seconde et le minimum de mémoire exigé pour le décodeur.

Un groupe d'images (GOP - Group of Pictures) est un ensemble d'images dans un ordre contigu. Il doit contenir au minimum une I-image. Un GOP peut commencer soit avec une B-image ou I-image et doit finir soit avec une I-image ou P-image. Si la

première image du groupe n'est pas dépendante des images dans le groupe précédent, alors ce GOP peut être codé et affiché indépendamment des autres GOPs.

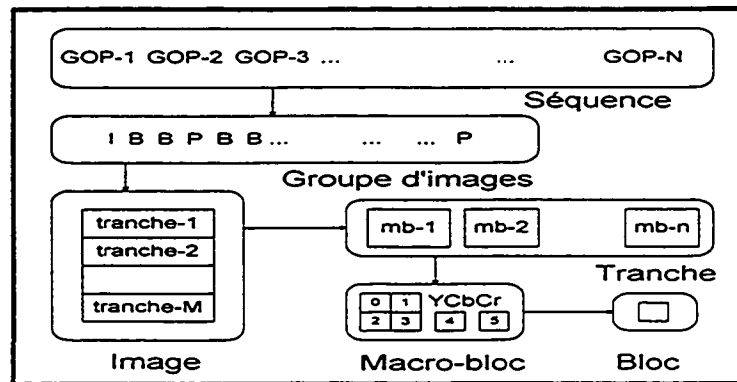


Figure 3.3 : Hiérarchie dans le système de codage

La troisième couche, qui est la couche d'image, définit l'information de codage pour chaque image. Les données incluses dans l'entête sont celles concernant le type d'image, la résolution et le choix des vecteurs de mouvement. Chaque image est divisée en plusieurs tranches. Ces tranches peuvent être grandes comme une image entière ou petites comme un seul macro-bloc. Dans le cas de corruption, l'information comprise dans l'entête de tranche permet au décodeur de récupérer les données et corriger les erreurs. En effet, le nombre de tranches par ligne dépend du niveau de bruit. Si le milieu est peu bruité, une tranche par image s'avère acceptable. Par contre, il est recommandé de couper une ligne en plusieurs tranches lorsque le milieu est considérablement perturbé.

Chaque tranche, quant à elle, est constituée de plusieurs macro-blocs. La tête du macro-bloc contient son type, ses coordonnées à l'intérieur de la tranche, l'information sur les vecteurs de mouvement et aussi l'information qui indique quel bloc du macro-bloc est actuellement codé et transmis.

Le bloc est l'unité de base pour l'encodage. Chaque bloc contient 64 pixels arrangés en une matrice de 8x8. Il convient de noter que les pixels ne sont pas encodés individuellement mais plutôt par bloc.

3.1.3 Techniques de compression MPEG-2

Les standards MPEG-2 et MPEG-1 sont fondés sur deux techniques de compression de base : la compensation de mouvement et le codage DCT (Discrete Cosine Transform).

En effet, la compensation de mouvement est très efficace pour la réduction des redondances temporelles alors que le codage DCT est utilisé dans plusieurs techniques pour réduire les redondances spatiales et perceptuelles. Les techniques de compensation de mouvement sont appliquées avec deux prédicteurs : causal (codage purement prédictif) et non-causal (codage interpolé).

L'erreur de la prédiction par le mouvement est ensuite compressée avec la réduction des redondances spatiales (DCT). Les données concernant le mouvement sont basées sur des blocs 16x16 et sont transmises ensemble avec les données spatiales. Enfin, pour obtenir un maximum d'efficacité, les données de mouvement sont compressées en utilisant le codage à longueur variable.

3.1.3.1 Réduction des redondances temporelles

a) Codage prédictif temporel

Cette technique de codage augmente le taux de compression en se basant sur la structure DPCM (Differential Pulse Code Modulation). Ayant la valeur des pixels décodés antérieurement et grâce à la compensation de mouvement, l'encodeur et le décodeur peuvent estimer ou prédire la valeur des pixels à être encodés ou décodés. La différence entre la valeur prédite et la valeur réelle sera encodée et transmise. Au récepteur, avec le signal de différence, le décodeur corrige l'erreur de la prédiction.

L'avantage de cette technique est que le signal de différence est souvent petit et se situe près de zéro. En effet, il est mieux compressé que les valeurs des pixels elles-mêmes qui ont normalement une variance beaucoup plus grande.

b) Compensation de mouvement

Pour obtenir une meilleure prédiction temporelle, le standard suggère l'utilisation de la technique d'estimation de mouvement. Il s'agit de calculer les vecteurs de déplacement associés à chaque macro-bloc. Le mouvement est modélisé ainsi par un vecteur bidimensionnel déplaçant un macro-bloc à un nouvel endroit. Grâce à ces vecteurs, le système (encodeur et décodeur) prédit les pixels d'un macro-bloc en utilisant la valeur des pixels du macro-bloc de l'image décodée (précédente ou future). Le macro-bloc déplacé suivant les vecteurs de mouvement devient le macro-bloc prédit de l'image à encoder. La figure 3.4 illustre le processus de la compensation de mouvement bidirectionnelle [2].

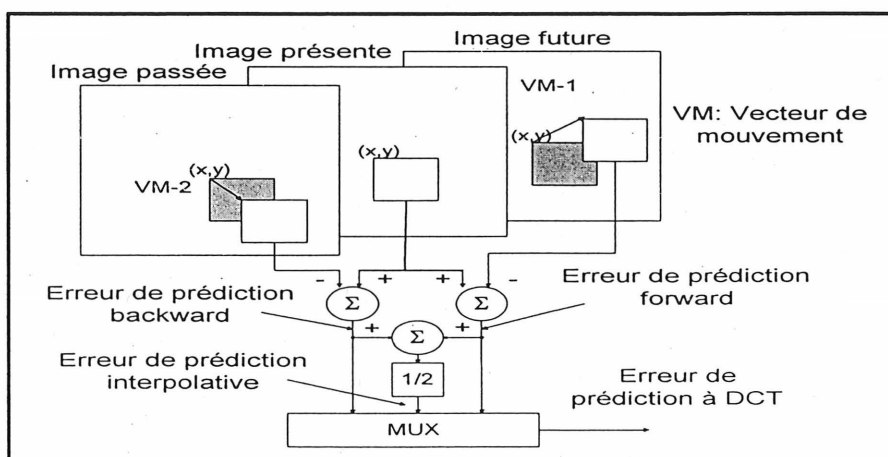


Figure 3.4 : Compensation de mouvement bidirectionnelle

c) Interpolation d'image

La compensation de mouvement peut être de type prédiction (P-image) ou de type bidirectionnelle (B-image). Dans le cas de P-image, on n'a qu'à déplacer le macro-bloc suivant le vecteur de mouvement pour obtenir l'image prédite. Cependant, pour les B-images, la compensation de mouvement est basée en même temps sur l'image passée et sur l'image future, et alors l'image prédite peut être formée en utilisant la technique d'interpolation des macro-blocs déplacés suivant les vecteurs de mouvement calculés. La différence entre le bloc prédit et le bloc original est ensuite transformée par DCT.

3.1.3.2 Réduction des redondances spatiales

a) Transformation DCT (Discrete Cosine Transform)

De manière intuitive, la transformation DCT peut être considérée comme un analyseur de fréquence, et la transformation DCT inverse comme un synthétiseur de fréquence. L'opérateur DCT convertit un bloc de 8x8 pixels de l'image en 64 coefficients. Chacun de ces derniers contient des informations fréquentielles du signal à l'entrée. Ces coefficients sont uniques, c'est-à-dire que pour un bloc donné, il existe un et un seul ensemble de coefficients correspondant à celui-ci.

Mathématiquement, la transformée DCT est réversible. L'information originale peut être reconstituée parfaitement. Mais dans le standard MPEG-2, pour réduire le débit de transmission, les coefficients de la transformée DCT sont soumis à une quantification, qui est une opération irréversible. Mathématiquement, la transformée se définit comme suit :

$$F(u, v) = \frac{1}{4} C(u) C(v) \sum_{x=0}^7 \sum_{y=0}^7 f(x, y) \cos(\pi(2x+1)u/16) \cos(\pi(2y+1)v/16) \quad (3.1)$$

avec

$$C(u) = \begin{cases} \frac{1}{\sqrt{2}} & \text{pour } u = 0 \\ 1 & \text{ailleurs} \end{cases} \quad C(v) = \begin{cases} \frac{1}{\sqrt{2}} & \text{pour } v = 0 \\ 1 & \text{ailleurs} \end{cases}$$

où

(x,y) est la coordonnée spatiale,

(u,v) est la coordonnée dans le domaine fréquentielle,

$x, y, u, v = 0, 1, 2, \dots, 7$.

La figure 3.5 illustre la transformation des pixels en coefficients DCT. En général, l'énergie se concentre majoritairement sur les coefficients de basse fréquence (DCT) situés sur le coin supérieur gauche de la matrice des coefficients. Le coefficient localisé à $(0,0)$ du bloc représente la fréquence 0 horizontalement et 0 verticalement, et est appelé le coefficient DC. Ce coefficient est proportionnel à la moyenne des valeurs des blocs.

Étant donné que la moyenne des pixels varie peu d'un bloc à un autre du voisinage, l'emploi du codage prédictif est idéal pour augmenter le gain de compression. Les autres coefficients sont appelés les coefficients AC. Ceux-ci varient avec le contenu du bloc considéré. Par conséquent, d'autres traitements plus complexes, dont la quantification, y sont appliqués.

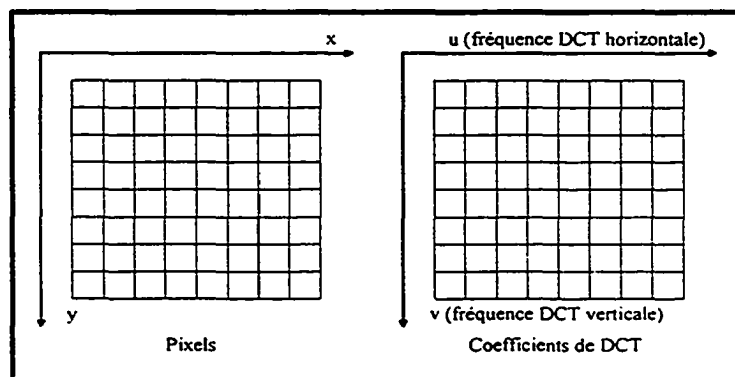


Figure 3.5 : Transformation des pixels en coefficients DCT

b) Quantification

La quantification permet de réduire le nombre de bits à transmettre en limitant le nombre de niveaux permis. En effet, chaque niveau quantifié représente un intervalle de valeurs par la valeur nominale de ce même intervalle. La conversion d'un chiffre réel en un chiffre entier est une forme de quantification scalaire. Ainsi, les coefficients DCT sont quantifiés scalairement suivant la formule suivante :

$$C_q(i, j) = \left[\frac{C(i, j)}{q(i, j)} \right] q(i, j) \quad (3.2)$$

où $\lceil . \rceil$ désigne la partie entière, $C(i, j)$ le coefficient DCT non-quantifié et $C_q(i, j)$ le coefficient DCT quantifié. Les pas de quantification $q(i, j)$ sont regroupés dans des matrices de quantification. La différence entre la valeur originale et la valeur quantifiée est appelée l'erreur de quantification. Une telle erreur est souvent imperceptible à l'œil humain, ce qui rend l'algorithme d'encodage plus efficace.

Similaire au MPEG-1, le codec MPEG-2 utilise à la fois le codage *inter-image* et le codage *intra-image* en appliquant deux matrices de quantification. Mais, les matrices de quantification dans le codec MPEG-1 peuvent être changées seulement au niveau de séquence tandis que dans le cas de MPEG-2 elles peuvent être changées au niveau d'image de la hiérarchie. La matrice donnée par l'équation 3.2 est employée dans le codage *intra-image*, et la matrice donnée par l'équation 3.3 dans le codage *inter-image*.

$$m_1(u, v) = \begin{bmatrix} 8 & 16 & 19 & 22 & 26 & 27 & 29 & 34 \\ 16 & 16 & 22 & 24 & 27 & 29 & 34 & 37 \\ 19 & 22 & 26 & 27 & 29 & 34 & 34 & 38 \\ 22 & 22 & 26 & 27 & 29 & 34 & 37 & 40 \\ 22 & 26 & 27 & 29 & 32 & 35 & 40 & 48 \\ 26 & 27 & 29 & 32 & 35 & 40 & 48 & 58 \\ 26 & 27 & 29 & 34 & 38 & 46 & 56 & 69 \\ 27 & 29 & 35 & 38 & 46 & 56 & 69 & 83 \end{bmatrix} \quad (3.3)$$

$$m_2(u, v) = \begin{bmatrix} 16 & 16 & 16 & 16 & 16 & 16 & 16 & 16 \\ 16 & 16 & 16 & 16 & 16 & 16 & 16 & 16 \\ 16 & 16 & 16 & 16 & 16 & 16 & 16 & 16 \\ 16 & 16 & 16 & 16 & 16 & 16 & 16 & 16 \\ 16 & 16 & 16 & 16 & 16 & 16 & 16 & 16 \\ 16 & 16 & 16 & 16 & 16 & 16 & 16 & 16 \\ 16 & 16 & 16 & 16 & 16 & 16 & 16 & 16 \\ 16 & 16 & 16 & 16 & 16 & 16 & 16 & 16 \end{bmatrix} \quad (3.4)$$

Chaque bloc 8x8 de coefficients DCT est quantifié et ce, non uniformément. Le coefficient DC est quantifié de façon fixe, tandis que les coefficients AC utilisent plutôt la quantification avec l'affectation du facteur d'ajustement. En effet, le pas du quantificateur est déterminé par la matrice de quantification multipliée avec le facteur d'ajustement choisi. Il peut donc varier pour différents macro-blocs. Les coefficients quantifiés sont calculés par l'équation 3.4 :

$$i(u, v) = \frac{8 * c(u, v)}{q * m(u, v)} \quad (3.5)$$

où

$i(u, v)$ est le coefficient quantifié,

$c(u, v)$ est le coefficient original correspondant,

q est le facteur d'ajustement,

$m(u, v)$ est le pas de quantification associé au coefficient $c(u, v)$.

c) Codage à longueur variable

Cette technique de codage se compose de deux parties: le codage de parcours (*run-length*) et le codage de Huffman. Le codeur de parcours balaye les blocs des coefficients DCT quantifiés suivant le parcours en zigzag pour compter le nombre de zéros qui précèdent un coefficient non nul. Le codeur forme ensuite une paire avec le nombre de zéros comptés et le nombre de bits nécessaires pour coder le coefficient non

nul rencontré. Cette paire (nombre de zéros / nombre de bits) est alors encodée par le codage de Huffman.

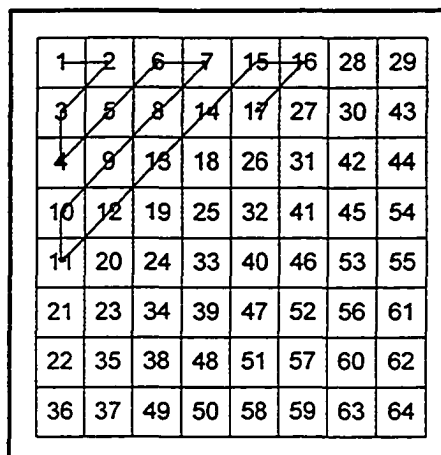


Figure 3.6 : Parcours en zigzag

Le codage de Huffman consiste à affecter à chaque paire un mot de code. Plus la fréquence d'apparition de la paire est grande, plus le mot de code affecté à celle-ci est court, et inversement, celles qui sont peu fréquentes sont associées à un plus long mot de code. En fait, le taux de compression varie proportionnellement avec la répétition des paires à l'entrée. Le tableau 3.1 suivant donne un exemple de l'utilisation du codage à longueur variable pour les coefficients AC quantifiés.

Tableau 3.1 : Codes de longueur variable pour les coefficients AC quantifiés

Nombre de zéros	Valeur	Code de longueur variable
1	1	011s
1	2	0001 10s
1	3	0010 0101s
1	4	0000 0011 00s
1	5	0000 0001 1011s

3.1.3.3 Système d'encodage

Le standard MPEG-2 spécifie la syntaxe et la sémantique de la séquence codée, ainsi que le travail effectué par le décodeur. Par contre, il ne spécifie pas exactement l'opération que doit faire l'encodeur. Ceci offre un très grand nombre de compromis entre le coût, la vitesse versus la qualité d'image et le taux de compression lors de la réalisation matérielle ou logicielle de l'encodeur.

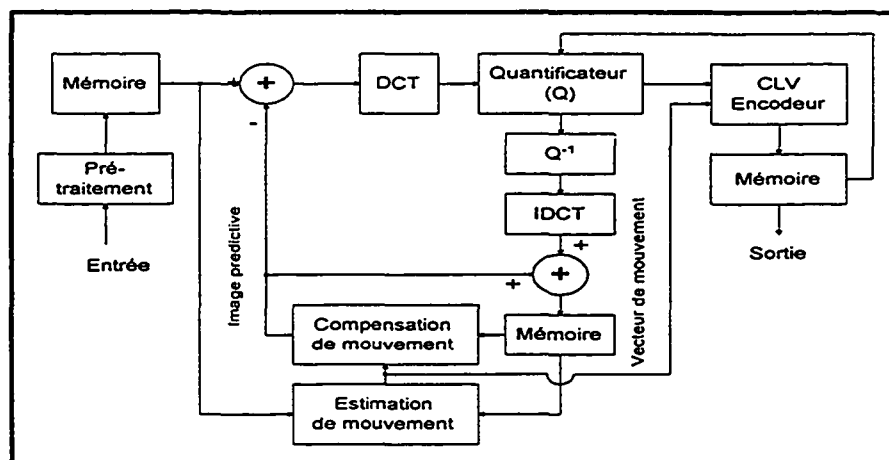


Figure 3.7 : Schéma-bloc de l'encodeur

La figure 3.7 présente le schéma-bloc de l'encodeur [19]. L'image source à l'entrée peut être progressive ou entrelacée. *A priori*, l'encodeur affecte un type à chaque image d'entrée (I-image, P-image, ou B-image). Selon le type de l'image, l'encodeur estime les vecteurs de mouvement pour chaque macro-bloc de l'image. Une P-image nécessite un seul vecteur de mouvement, alors qu'il en faut deux pour les B-images. Considérons la séquence d'images illustrée à la figure 3.8 a) comme l'entrée de l'encodeur.

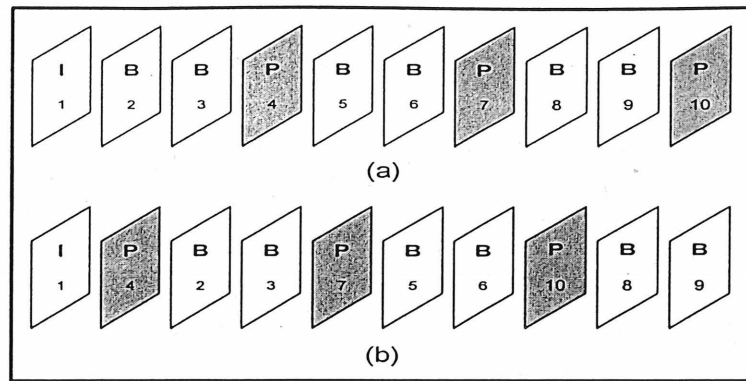


Figure 3.8 : Arrangement des images dans la séquence pour la compensation de mouvement.

- a) Une séquence typique à l'encodeur
- b) La même séquence après l'arrangement effectué par l'encodeur

Comme une B-image est encodée de façon bidirectionnelle, c'est-à-dire qu'elle utilise la compensation de mouvement basée en même temps sur l'image passée et sur l'image future, l'encodeur arrange les images dans la séquence de sorte que l'image future soit encodée avant l'encodage de l'image actuelle (Figure 3.8 b). Le décodeur s'occupe de faire l'arrangement inverse pour retrouver l'ordre initial des images dans la séquence.

L'unité de base pour la compensation de mouvement est le macro-bloc. L'image est donc décomposée en plusieurs macro-blocs. Les macro-blocs sont encodés séquentiellement de gauche à droite et de haut en bas.

Pour un macro-bloc donné, l'encodeur identifie en premier lieu son type de codage spécifique qui peut être, soit l'utilisation des vecteurs de mouvement calculés ou non, soit l'encodage de type *intra-image* ou de type *inter-image*, soit l'adaptation du facteur de quantification ou l'utilisation de son ancienne valeur. Le choix du type de codage repose sur l'efficacité de la compensation de mouvement dans la région considérée et sur le contenu du macro-bloc. Selon le type de codage choisi, l'encodeur

prédit le contenu du macro-bloc en utilisant la compensation de mouvement basée sur l'image passée ou sur l'image future.

Il calcule par la suite la différence entre le macro-bloc prédit et le vrai macro-bloc d'entrée. Le signal de différence est décomposé en des blocs de 8x8 pixels avant d'être transformés en des coefficients DCT. Ces coefficients sont ensuite quantifiés et transformés d'un bloc de dimension deux en un vecteur de dimension unitaire suivant le parcours en zigzag. En même temps, les informations utiles (type de macro-bloc, vecteur de mouvement) sont aussi encodées.

Finalement, l'algorithme de codage à longueur variable (CLV) est utilisé afin d'augmenter le gain de compression. La séquence binaire est maintenant formée au complet. Il reste à décoder les I-images et les P-images pour les utiliser comme référence lors de la prédiction des macro-blocs des images suivantes. Le processus se répète ainsi pour encoder chacune des images.

3.1.3.4 Système de décodage

Le décodage est l'inverse du processus d'encodage. À la figure 3.9, nous présentons le schéma-bloc simplifié d'un décodeur.

Pour faciliter la compréhension, considérons de nouveau la séquence illustrée à la figure 3.8 comme l'entrée de l'encodeur. La séquence codée arrive et s'accumule dans la mémoire jusqu'au moment où le décodeur commence à faire du décodage. Le décodeur de longueur variable décode l'entête de la première image, de numéro 1 et de type I, ainsi que les coefficients quantifiés. Ces coefficients passent à travers le module de quantification inverse pour redevenir les vrais coefficients de DCT, qui sont ensuite transformés inverse DCT afin de reconstituer les informations de l'image. L'image reconstruite est mise en mémoire (image passée).

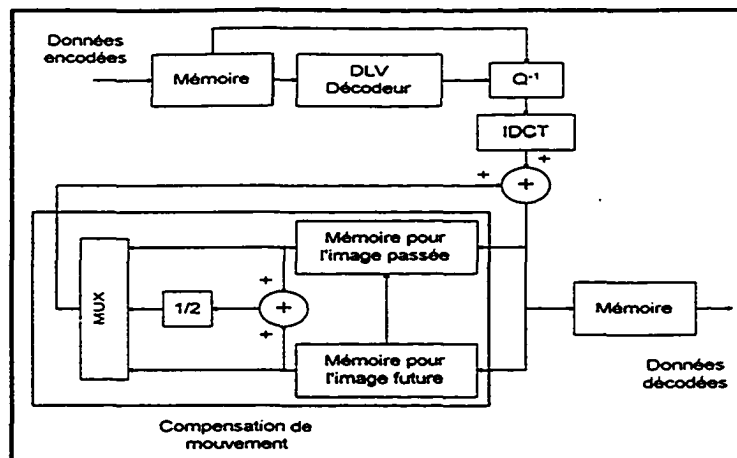


Figure 3.9 : Schéma-bloc du décodeur

Le décodeur continue à décoder l'image suivante, de numéro 4 et de type P. Pour chaque bloc, le décodeur DLV (Décodeur de longueur variable) décode les vecteurs de mouvement référencés à l'image sauvegardée de numéro 1, et les coefficients quantifiés. Ces coefficients sont par la suite quantifiés inverses et transformés DCT inverse pour devenir le signal de différence. Le décodeur effectue en parallèle la compensation de mouvement en appliquant les vecteurs de mouvement à l'image sauvegardée. L'addition du signal de différence avec l'image prédite donne l'image reconstruite. Elle est ensuite mise en mémoire (image future). Cette image ne sera pas visualisée jusqu'à ce que les images numéro 2 et 3 soient reçues, décodées et visualisées.

Le DLV continue à décoder l'image suivante, de numéro 2 et de type B. Il décode, pour chaque bloc, les vecteurs de mouvement référencés à l'image précédente et à l'image future, ainsi que les coefficients quantifiés. Ces derniers suivent les mêmes traitement que ceux de l'image de numéro 3 pour donner le signal de différence. Ce signal est additionné aux blocs prédits pour constituer l'image reconstruite. Le processus continue et on retrouve toute la séquence au décodeur.

3.2 Artefacts causés par le codec MPEG-2

Comme nous l'avons mentionné, quand le taux de compression devient plus élevé le standard commence à produire des erreurs qui réduisent la qualité visuelle des images vidéo. Ces erreurs causées par le codec se présentent sous plusieurs formes selon le contenu des images et le débit de transmission.

La figure 3.10 illustre le résultat du codage MPEG-2 à un débit de 3 Mbit/s. Plusieurs défauts deviennent subjectivement discernables. En général, il y a trois principaux types de dégradations sur les images décodées par le standard MPEG-2 : l'effet de bloc, le bruit de moustique et le bruit de quantification [8,15,25,28,31,34].

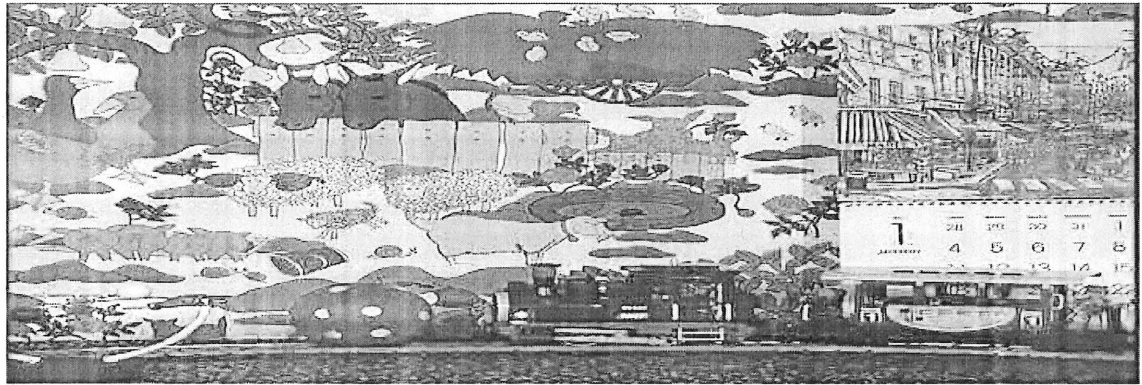
3.2.1 Effet de bloc

Cette erreur et sa réduction ont été le sujet le plus souvent abordé dans le domaine du post-traitement des images JPEG (Joint Photographic Expert Group) et MPEG (Moving Pictures Experts Group).

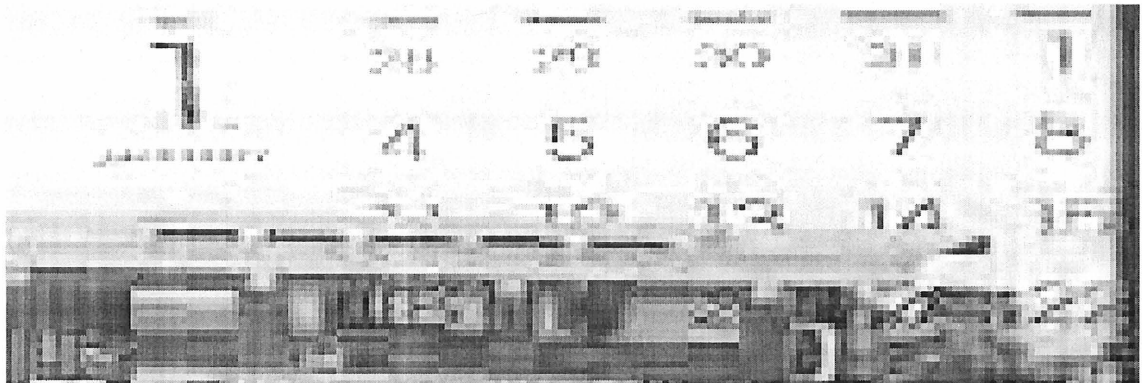
En regardant une image complète, on peut remarquer qu'il a des zones constituées de blocs indépendants. Cette dégradation est produite parce que, dans ces standards, chaque bloc de l'image est encodé (DCT) séparément. À l'intérieur d'un bloc 8x8, les premiers coefficients de la DCT détermine généralement la surface « moyenne » du bloc. Cela introduit, surtout quand le débit de transmission est bas, une grande erreur de quantification et mène à une perte de continuité entre les surfaces des blocs reconstruits.

Cette erreur se manifeste sous la forme d'une grande variation des valeurs des pixels aux extrémités des blocs voisins. L'œil interprète cette variation comme un contour. Ce phénomène est appelé l'effet de bloc. En général, cet effet survient dans

tous les systèmes de quantification de type « block-based » lorsqu'ils fonctionnent à bas débit.



a)



b)

Figure 3.10 : Image décodée par MPEG-2 à un débit de 3Mbit/s

a) Image « Mobile »

b) Agrandissement de l'image « Mobile »

3.2.2 Bruit de moustique

En regardant les matrices de quantification (équations 3.3 et 3.4), on peut constater que le pas de quantification augmente suivant le parcours en zigzag défini par le standard. Sous un autre angle, on voit que les coefficients représentant les fréquences diagonales sont encodés avec moins de précision par rapport aux autres fréquences car on a présumé que la probabilité que ces fréquences surviennent est petite. En effet, lorsque l'image contient suffisamment de ces fréquences, le codec ne permet pas de les

coder avec une précision adéquate. La sortie, par conséquent, contient beaucoup d'erreur sur les contours diagonaux.

Cet artefact est nommé le bruit de moustique ou « mosquito noise ». Ce bruit introduit, dans une séquence d'images animées, l'effet de papillotement. En général, cette erreur est évidente dans les zones plates contenant un contour faible. Par contre, dans les zones de haute fréquence, ce bruit est noyé dans le contenu de l'image et devient donc moins visible.

3.2.3 Bruit de quantification

Le bruit de quantification est un bruit aléatoire. Il est produit par le processus de quantification. Il y a deux bruits qui apparaissent dans ce processus. Le premier est l'erreur à haute fréquence qui est causée par l'élimination des coefficients AC à haute fréquence. L'autre est l'erreur de quantification qui est produite lorsqu'on augmente le pas de quantification pour obtenir un taux de compression plus haut. Puisque la transformation DCT dans le standard MPEG-2 est exécutée bloc par bloc, ce bruit de quantification dépend aussi de chaque bloc dans l'image.

3.3 *Post-traitement*

Après les étapes encodage/décodage, le post-traitement, quant à lui, est un processus pour enlever les défauts qui sont produits par les techniques de compression. Plusieurs techniques de post-traitement ont été proposées. Certaines s'attaquent à un défaut spécifique, d'autres sur l'ensemble des défauts dans l'image. Mais jusqu'à présent, la plupart des algorithmes ont été développés pour le traitement des effets de bloc dans les images JPEG [10,27,28,31,35]. Par contre dans les images MPEG-2, c'est le bruit de moustique qui dégrade le plus la qualité visuelle. Puisque la plupart de ce bruit se trouve dans les zones autour des contours, nous utilisons les réducteurs présentés dans la section 2.2 pour concevoir notre système de post-traitement pour le

signal vidéo MPEG-2 afin d'améliorer la qualité visuelle des images.

3.3.1 Traitement de la luminance Y

Pour la luminance, notre système de post-traitement se compose de trois grandes parties. Le diagramme dans la figure 3.11 illustre la structure de ce système.

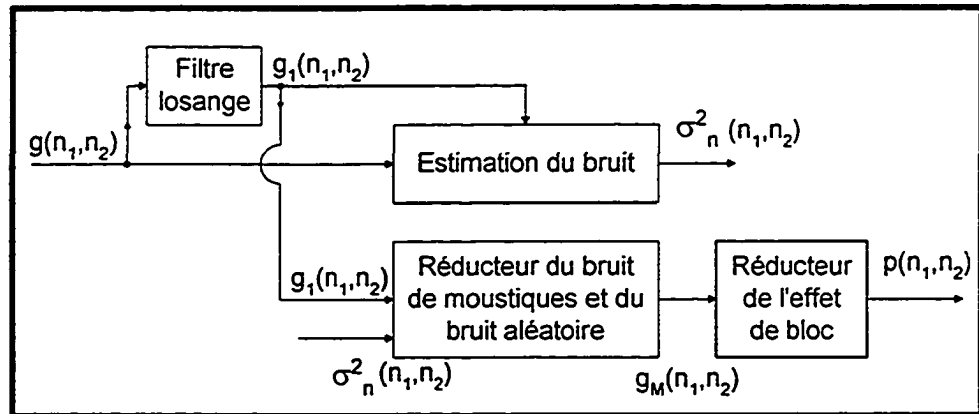


Figure 3.11 : Système du post-traitement de MPEG-2 pour la luminance Y

Dans tous les systèmes de réduction de bruit qui ont été présentés dans le chapitre précédent, nous avons supposé que les bruits des images sont bien connus d'avance. Dans le cas du post-traitement de MPEG-2, cette information n'est pas disponible.

Alors, la première étape de ce système consiste à estimer la puissance de bruit de l'image dégradée. Puis, en utilisant cette information estimée et l'un des réducteurs présentés dans la section 2.2 notre système réduit le bruit de moustique et le bruit de quantification. La dernière étape sert à réduire l'effet de bloc.

Tout d'abord, un filtre spatial de forme d'un losange est appliqué. Nous utilisons ce type de filtrage pour estimer la puissance de bruit et pour réduire une certaine quantité de l'effet de bloc et du bruit aléatoire. Dans les images vidéo de MPEG-2, ces types de bruit sont généralement à haute fréquence spatiale (verticale et horizontale). De

plus, la résolution diagonale de l'œil humain est plus faible que celles horizontale et verticale, et il est donc important de conserver autant que possible les résolutions horizontale et verticale de l'image originale. La forme d'un losange dans le domaine fréquentiel paraît donc appropriée à ces spécifications.

L'équation 3.5 et la figure 3.12 présentent la réponse impulsionnelle aussi bien que la réponse fréquentielle du filtre en losange utilisé dans ce système. En effet, ce filtre est de type passe-bas dont la bande passante est à 75% de la moitié de la fréquence d'échantillonnage. Il est conçu en utilisant les techniques de transformation de McClellan [11].

$$h_{\text{losange}}(n_1, n_2) = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 3 & -8 & 3 & 0 & 0 \\ 0 & 3 & -16 & 25 & -16 & 3 & 0 \\ 1 & -8 & 25 & 224 & 25 & -8 & 1 \\ 0 & 3 & -16 & 25 & -16 & 3 & 0 \\ 0 & 0 & 3 & -8 & 3 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} / 256 \quad (3.6)$$

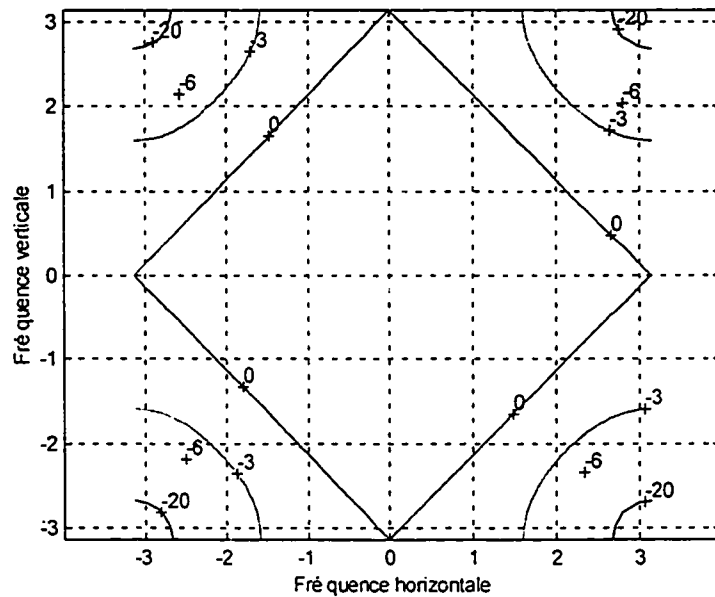


Figure 3.12 : Réponse fréquentielle du filtre en losange à 75%

3.3.1.1 Estimation de la puissance de bruit

Puisque la plupart de la puissance de bruit se trouve à haute fréquence, la différence $n(n_1, n_2)$ entre l'image dégradée $g(n_1, n_2)$ et l'image filtrée $g_1(n_1, n_2)$ par le filtre en losange $h_{losange}(n_1, n_2)$ peut être considérée comme le bruit qui est présent dans l'image. En utilisant cette image de bruit et le principe de la segmentation discuté dans la section 2.2.1, nous créons l'image $\sigma(n_1, n_2)$ des écarts types locaux de chaque pixel de l'image $n(n_1, n_2)$ selon la dimension de la fenêtre de traitement et de la valeur de seuil S précise. La structure de ce système d'estimation de la puissance du bruit peut être illustrée comme le montre la figure 3.13 :

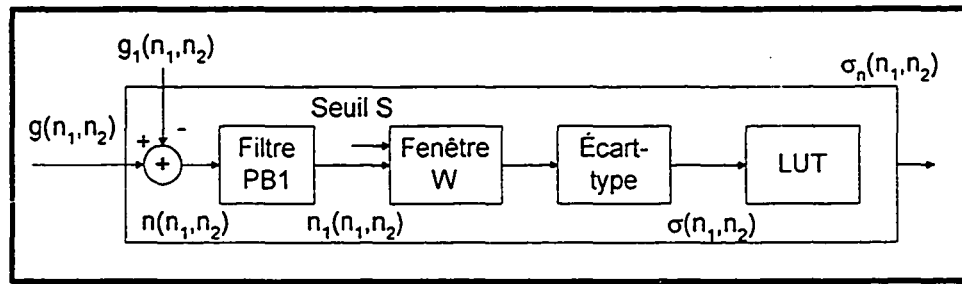


Figure 3.13 : Système d'estimation du bruit

Le filtre passe-bas utilisé ici est un filtre moyen de dimension 3x3 dont les coefficients sont égaux à ceux du filtre $h_1(n_1, n_2)$ présenté à la figure 2.2. La façon de calculer l'image des écarts-types locaux est la suivante :

$$\sigma(n_1, n_2) = \frac{1}{N} \sum \left| n_1(n_1, n_2) - \hat{m}_n(n_1, n_2) \right| \quad (3.7)$$

avec

$$\hat{m}_n(n_1, n_2) = \frac{1}{N} n_1(n_1, n_2) \quad (3.8)$$

où N est le nombre des pixels dont les valeurs absolues entre eux et le pixel central sont plus petites que la valeur de seuil S précise. À partir de l'image des écarts-types locaux $\sigma(n_1, n_2)$, la variance du bruit pour chaque pixel de l'image décodée peut être calculée.

Nous avons déterminé empiriquement une table de référence (LUT) pour déterminer la variance du bruit. La corrélation entre l'écart-type local $\sigma(n_1, n_2)$ et l'écart-type du bruit estimé $\sigma_n(n_1, n_2)$ est représentée par la courbe illustrée à la figure 3.14. Ainsi, lorsque l'écart-type du bruit de l'image est bien estimée, nous pouvons utiliser les réducteurs de bruit proposés à la section 2.2 pour réduire le bruit de moustique et le bruit aléatoire de l'image MPEG-2.

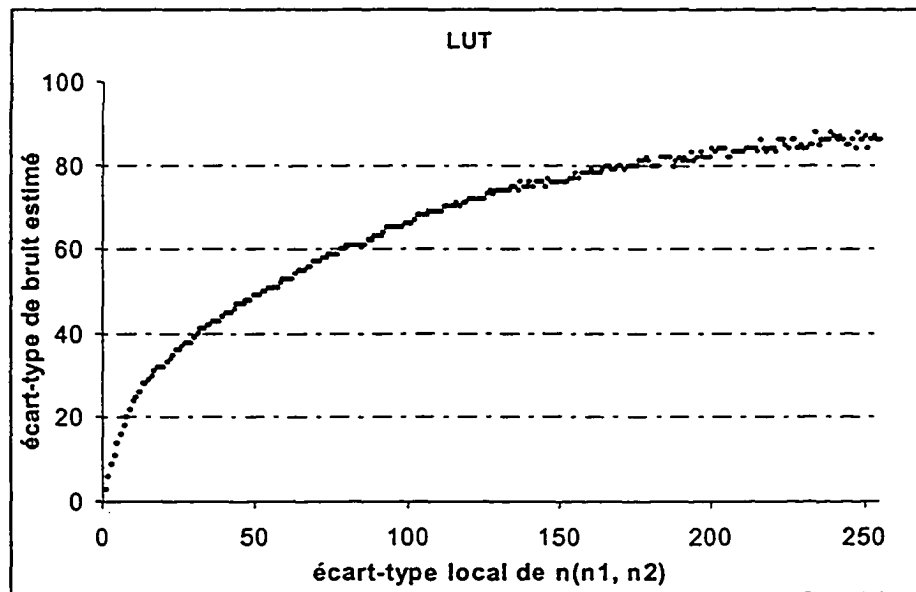


Figure 3.14 : Courbe d'estimation d'écart-type du bruit

3.3.1.2 Réduction du bruit de moustique et du bruit aléatoire

Comme nous l'avons mentionné, ce qui dégrade le plus la qualité visuelle des images MPEG-2 est le bruit de moustique. Jusqu'à présent, il n'y a pas beaucoup de travaux qui visent à réduire ce type de bruit [8,17,34]. En fait, la majeure partie de ce bruit se trouve autour des contours de l'image décodée.

À la section 2.2, nous avons proposé quatre réducteurs de bruit différents qui sont capables de réduire le bruit autour des contours grâce à la segmentation : l'algorithme de segmentation, l'approche utilisant la moyenne des images, l'approche

utilisant une image constante et l'approche combinée. En effet, nous pouvons appliquer chacun de ces réducteurs dans notre système de post-traitement. Les performances des quatre réducteurs de bruit dans la section 2.2 dépendent de la quantité du bruit restée dans les zones plates de l'image à traiter. Dans le cas du standard MPEG-2, les zones plates des images décodées sont déjà lisses. Donc, les deux dernières approches ne sont pas significativement meilleures que les deux premières.

De plus, comme elles sont plus complexes, nous n'utilisons pas l'approche constante et l'approche combinée dans notre système de post-traitement pour retenir seulement l'algorithme de segmentation (section 2.2.1) et l'approche utilisant la moyenne des images (section 2.2.2). Alors, nous avons deux systèmes :

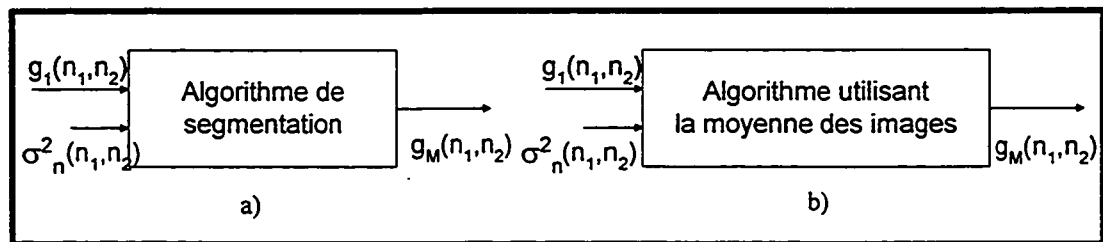


Figure 3.15 : Systèmes de réduction du bruit de moustique et du bruit aléatoire

- a) Utilisant l'algorithme de segmentation
- b) Utilisant l'algorithme de la moyenne des images

Selon ces systèmes, l'image d'entrée n'est plus l'image dégradée $g(n_1, n_2)$. Elle est l'image $g_1(n_1, n_2)$ filtrée par le filtre spatial en losange. L'autre entrée est la variance du bruit que nous avons estimée de la façon présentée à la section 3.3.1.1. L'image sortie $g_M(n_1, n_2)$ de ce système entre ensuite dans le système de réduction de l'effet de bloc comme entrée.

Nous présentons les résultats des images traitées par la technique de segmentation aussi que ceux obtenus par l'algorithme utilisant la moyenne des images à la section 3.3.3.

3.3.1.3 Réduction des effets de bloc

Après deux étapes du filtrage par le filtre en losange et par le filtre spatial adaptatif avec la configuration présentée à la section 3.3.1.2, on obtient l'image de sortie $g_M(n_1, n_2)$. Au sens de la qualité visuelle, cette image ne contient plus de bruit aléatoire et de bruit de moustique. Cependant, il reste toujours une quantité de bruit qui se présente sous la forme des effets de bloc. Dans cette section nous présentons la façon d'enlever ces défauts.

Pendant les dernières années, la réduction des effets de bloc a été le sujet le plus souvent abordé. Une méthode simple, c'est le filtrage linéaire qui permet d'enlever les composantes de haute fréquence situées sur les bordures des blocs. Cette technique réduit effectivement les faux contours causés par l'effet de bloc mais, en même temps, elle affecte la continuité des vrais contours dans le cas où ces derniers parcourent plusieurs blocs adjacents. Ramamurthi et Gersho [35] utilisent des filtres non-linéaires et spatialement variants pour tout pixel dans l'image tandis que Shu et Chen [30] emploient des filtres médians modifiés. Dans leur travail, Kuo et Hsieh [10] utilisent les techniques d'amincissement pour améliorer la performance des filtres sur les contours. Récemment, plusieurs méthodes fondées sur les projections dans des ensembles convexes sont proposées et permettent une bonne amélioration [18,27,32,33].

Dans notre projet, nous n'avons pas l'intention de comparer tous les algorithmes de réduction des effets de bloc. En nous basant toujours sur les principes des statistiques locales et de la segmentation présentés dans le chapitre 2, nous avons développé notre propre système pour éliminer ces défauts. Il est bien connu qu'il y a des zones constituées de bloc indépendants sur les images décodées. Ces erreurs se manifestent sous la forme d'une grande variation des valeurs des pixels aux extrémités des blocs voisins qui ont une dimension de 8×8 . Alors, nous pouvons localiser les défauts et concentrer notre correction uniquement sur les zones où les erreurs se manifestent.

L'algorithme proposé dans cette section se compose de deux grandes techniques essentielles. La première est l'utilisation des filtres en sous-bande pour décomposer l'image d'entrée en quatre bandes principales. La deuxième est la correction des défauts sur chaque image de sous-bande en exploitant les principes des statistiques locales, de la segmentation et de l'information sur la location des erreurs. Pour faciliter la compréhension, nous présentons la structure de notre système à la figure 3.16 :

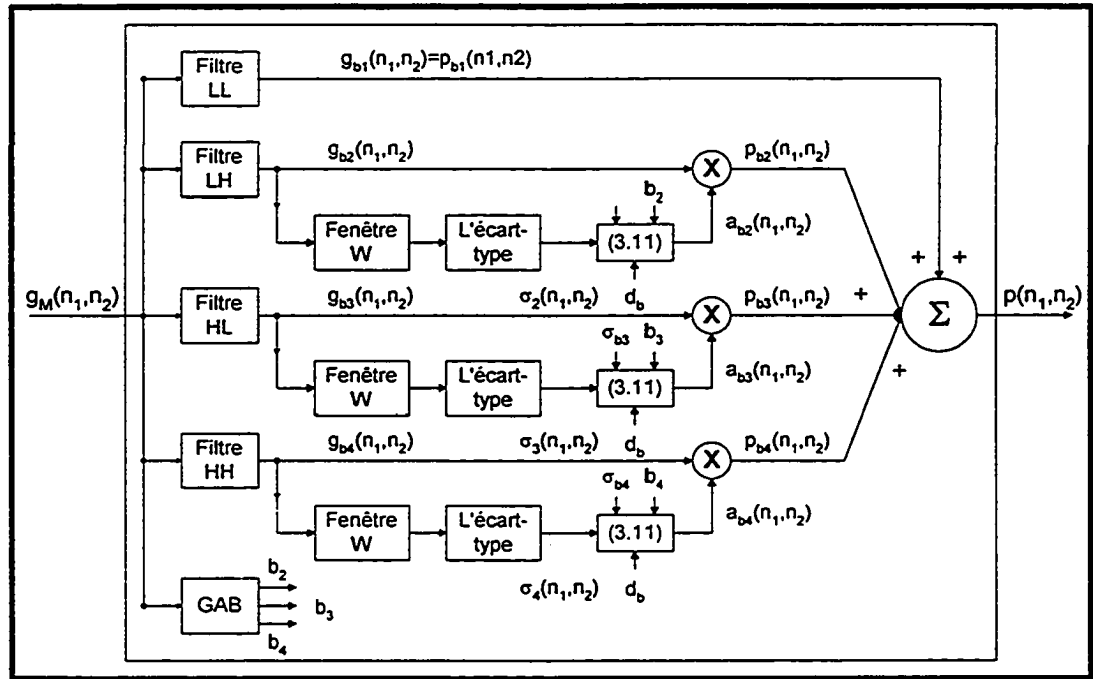


Figure 3.16 : Système de réduction des effets de bloc

Comme le montre la figure 3.16, l'image de sortie $g_M(n_1, n_2)$ du système de réduction du bruit de moustique entre dans le système de réduction des effets de bloc. Ensuite, elle est filtrée parallèlement par quatre filtres de sous-bande : bas-bas, bas-haut, haut-bas et haut-haut. L'image traitée par le filtre bas-bas ne contient que les composantes à basse fréquence de l'image décodée et nécessite aucun traitement supplémentaire. Dans les images filtrées par les filtres bas-haut, haut-bas et haut-haut, il y a les composants à haute fréquence présentant des effets de bloc sur les directions verticales, horizontales et diagonales respectivement. Les coefficients des filtres appliqués sont présentés à l'équation 3.9 :

$$\begin{aligned}
f_{LL}(n_1, n_2) &= \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} / 9 & f_{LH}(n_1, n_2) &= \begin{bmatrix} -1 & -1 & -1 \\ 2 & 2 & 2 \\ -1 & -1 & -1 \end{bmatrix} / 9 \\
f_{HL}(n_1, n_2) &= \begin{bmatrix} -1 & 2 & -1 \\ -1 & 2 & -1 \\ -1 & 2 & -1 \end{bmatrix} / 9 & f_{HH}(n_1, n_2) &= \begin{bmatrix} 1 & -2 & 1 \\ -2 & 4 & -2 \\ 1 & -2 & 1 \end{bmatrix} / 9
\end{aligned} \tag{3.9}$$

Pour corriger les effets de bloc dans les trois sous-images à haute fréquence nous utilisons encore les principes des statistiques locales et la segmentation mais, cette fois, avec une amélioration. Puisque nous connaissons bien où les défauts se trouvent, la correction est exécutée seulement sur les bordures des blocs de l'image. Ce filtrage permet d'enlever les erreurs de bloc tout en conservant les vrais contours et les détails dans les cas où ils parcourent plusieurs blocs adjacents. Afin de localiser les pixels à corriger dans chaque sous-image nous utilisons un générateur d'adresses de bloc (GAB) qui crée les images de référence dont les directions sont illustrées à la figure 3.17 :

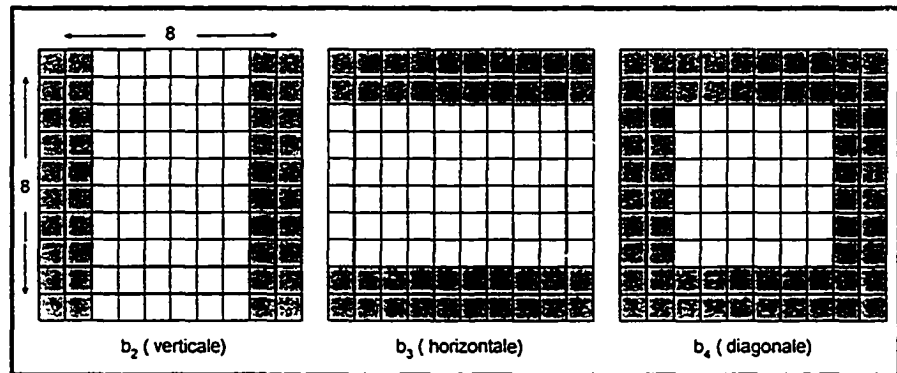


Figure 3.17 : Localiser les pixels à corriger dans les directions verticales, horizontales et diagonales.

Alors, dans les trois sous-images à haute fréquence (basse-haute, haute-basse et haute-haute) seuls les pixels situés aux emplacements de couleur grisée dans les images de référence de la figure 3.15 sont pris en compte pour le calcul. Les images de sortie $p_{bi}(n_1, n_2)$ pour chaque sous-bande sont déterminées avec ces équations :

$$p_{bi}(n_1, n_2) = a_{bi}(n_1, n_2)g_{bi}(n_1, n_2) \quad (3.10)$$

avec

$$a_{bi}(n_1, n_2) = \begin{cases} \frac{\sigma_i^2(n_1, n_2) - \sigma_{bi}^2}{\sigma_i^2(n_1, n_2)} & \text{pour } \sigma_i^2(n_1, n_2) > \sigma_{bi}^2 \\ 0 & \text{ailleurs} \end{cases} \quad (3.11)$$

où $\sigma_i^2(n_1, n_2)$ sont calculées de la même façon que celle présentée aux équations 3.6 et 3.7; σ_{bi}^2 sont des constantes fixées, dont la valeur dépend du débit de la séquence vidéo.

Avec les quatre sous-images de sortie $p_{bi}(n_1, n_2)$, nous obtenons l'image traitée $p(n_1, n_2)$ du système de réduction des effets de bloc à partir de l'équation 3.12, et ainsi de notre système de post-traitement.

$$p(n_1, n_2) = p_{b1}(n_1, n_2) + \sum_{i=2}^4 p_{bi}(n_1, n_2) \quad (3.12)$$

Quand nous regardons l'image traitée $p(n_1, n_2)$ à la sortie du système de correction de défauts, nous constatons qu'il y a une grande amélioration en matière de qualité visuelle. Les images de résultats sont présentées à la section 3.3.3. Ce système permet effectivement d'enlever les trois types de défauts causés par le codec qui sont les effets de bloc, le bruit de moustique et le bruit de quantification. L'image traitée devient nettement plus propre. Elle contient beaucoup moins de défauts.

3.3.2 Traitement des chrominances U et V

Jusqu'à présent, il n'y a aucun travail qui présente les résultats du post-traitement pour les images de chrominance. Dans cette section, nous présentons notre système de post-traitement appliqué pour améliorer la qualité des chrominances U et V des séquences vidéo MPEG-2.

Puisque les défauts contenus dans les chrominances U et V sont de même type que ceux dans la luminance Y, nous utilisons la même structure et les mêmes principes que nous avons présentés à la section 3.3.1. Cependant, l'image d'entrée dans les

modules de segmentation de ce système ne sont plus les images de chrominance mais l'image de luminance sous-échantillonnée horizontalement. Cette modification peut être illustrée comme le montre la figure 3.18 :

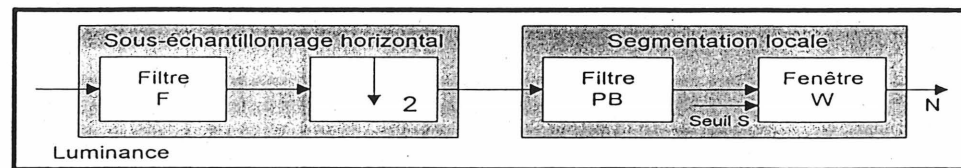


Figure 3.18 : Sous-échantillonnage horizontal de la luminance Y

Parce que l'image de luminance présente des informations plus importantes et plus détaillées que les images de chrominance, l'utilisation de l'image de luminance échantillonnée permet d'obtenir une segmentation plus précise, ce qui résulte en un meilleur résultat que l'utilisation des images de chrominance elle-même. Le sous-échantillonnage horizontal de l'image de la luminance Y est exécuté grâce à un filtre passe-bas F dont la réponse impulsionnelle et la réponse fréquentielle sont définies par l'équation 3.13 :

$$f(n) = [-5 \ 0 \ 37 \ 64 \ 37 \ 0 \ -5]/128 \quad (3.13)$$

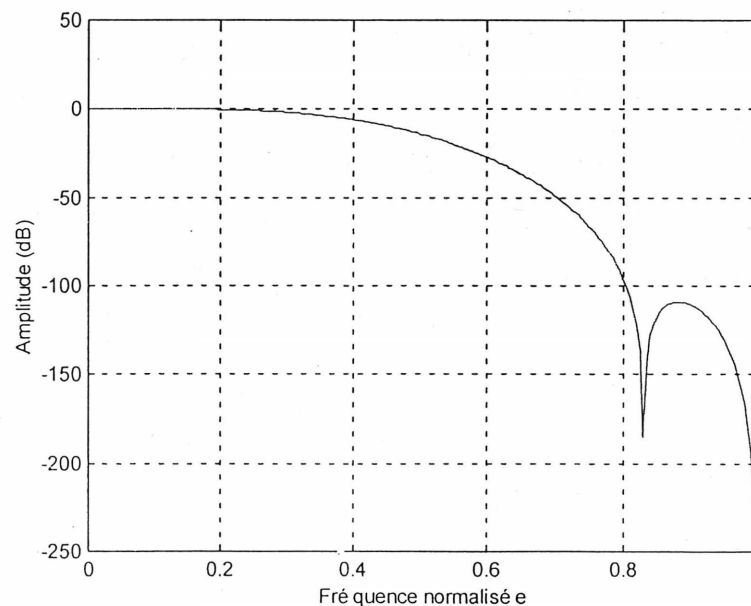


Figure 3.19 : Réponse fréquentielle du filtre passe-bas $F(\omega)$

Ce filtre est conçu en utilisant la technique de fenêtres. La fenêtre utilisée est celle de Kaiser [16]. Le filtre PB et la fenêtre W illustrée à la figure 3.18 sont similaires à ceux présentés dans l'algorithme de segmentation (section 2.2.1). À la sortie du système de post-traitement, les images de chrominance U et V deviennent plus « propres ». L'effet de bloc aussi bien que le bruit de moustique et le bruit de quantification sont réduits d'une façon significative. Nous obtenons les images traitées non seulement avec une meilleure qualité visuelle mais aussi avec un meilleur rapport signal sur bruit.

3.3.3 Présentation des résultats

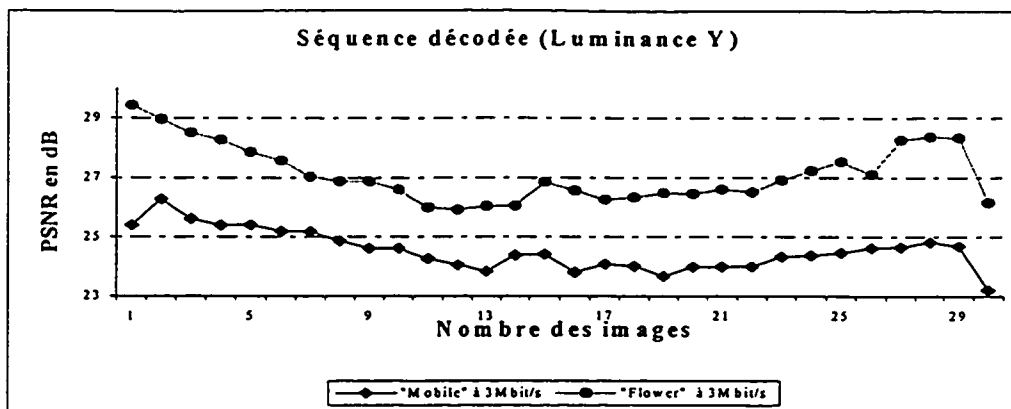
Cette section est réservée à la présentation des résultats des images traitées par notre système de post-traitement. Pour le réducteur de bruit de moustique et de bruit de quantification, nous avons utilisé deux techniques différentes : une fois avec l'algorithme de segmentation (section 2.2.1) et l'autre fois avec l'approche moyenne (section 2.2.2.1). Dans le cas utilisant l'algorithme de segmentation, la fenêtre de traitement a une dimension de 7×7 et la valeur de seuil S est égale à 12. Pour l'approche moyenne, nous avons utilisé des fenêtres de dimension 7×7 et 11×11 avec deux valeurs de seuil égales à 12 et 14.

Dans nos tests, nous avons utilisé deux séquences vidéo : « Mobile » et « Flower », décodées avec un débit de 3 à 8 Mbit/s. Comme illustré dans les figures 3.20 et 3.21 l'amélioration quantitative pour la luminance Y est négligeable. Le rapport signal sur bruit amélioré est environ 0.05 dB en moyenne pour toute la séquence. Par contre, la qualité visuelle est fortement améliorée. Si nous regardons les images de résultats présentées dans les figures 3.22 et 3.24, nous pouvons constater qu'elles sont plus naturelles et plus « propres ».

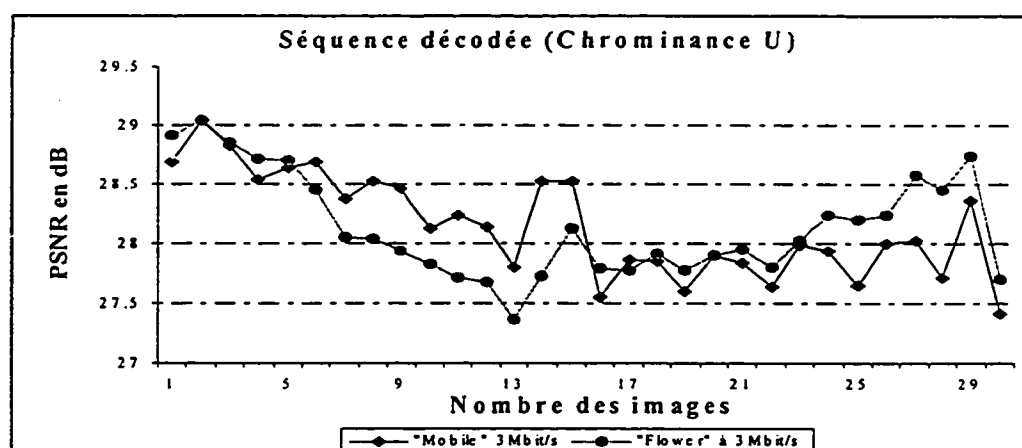
Ni l'effet de bloc, ni le bruit de moustique, ni le bruit aléatoire ne sont perceptibles dans les zones plates et dans les zones plates contenant des contours comme le calendrier de la séquence « Mobile » et le ciel de la séquence « Flower ». Cela permet une bonne qualité visuelle parce que l'œil humain est par nature sensible à tout bruit dans ces zones. Cependant, dans les zones de contours et de détails fins comme les agneaux de la séquence « Mobile » et les fleurs de la séquence « Flower », les erreurs ne sont pas beaucoup traitées parce qu'on présume qu'elles sont masquées par les détails de l'image et que l'œil humain les tolère.

Pour les images de la chrominance U et V, les résultats obtenus par le système de post-traitement sont très bons. La qualité des images est améliorée non seulement qualitativement mais aussi quantitativement. Nous avons gagné un rapport signal sur bruit amélioré jusqu'à 0.7 dB en moyenne pour la séquence « Mobile » et jusqu'à 0.5 dB pour la séquence « Flower » (Figures 3.20 et 3.21). Les améliorations quantitatives et qualitatives des composants U et V permettent d'obtenir les images de vidéo en couleur plus belles.

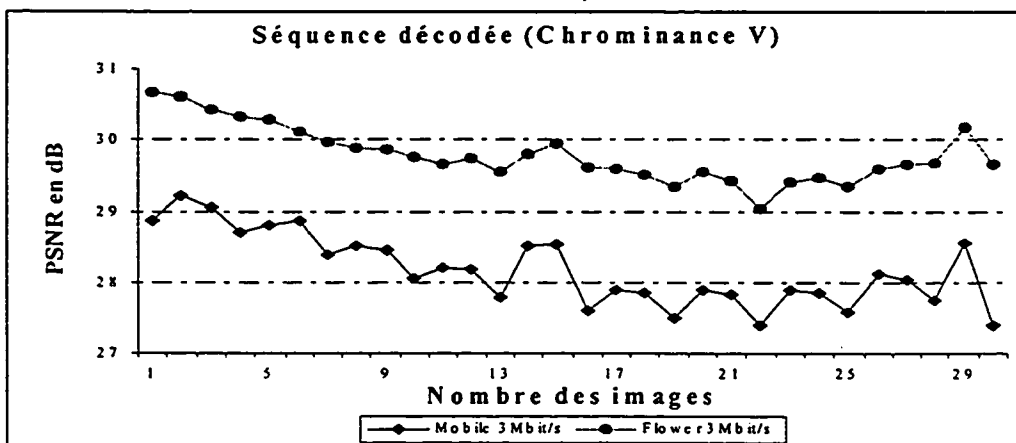
Dans les figure 3.20-3.25, nous présentons les rapports signal sur bruit améliorés et les images de résultats traitées par deux techniques : l'algorithme de segmentation et l'approche moyenne. Selon ces figures, nous voyons que l'approche moyenne donne légèrement une amélioration contre l'algorithme de segmentation. Son rapport signal sur bruit amélioré est plus haut et la qualité visuelle est aussi meilleure. Par contre, sa complexité est environs 4 fois plus grande que celle de l'algorithme de segmentation (section 2.3.4).



a)



b)



c)

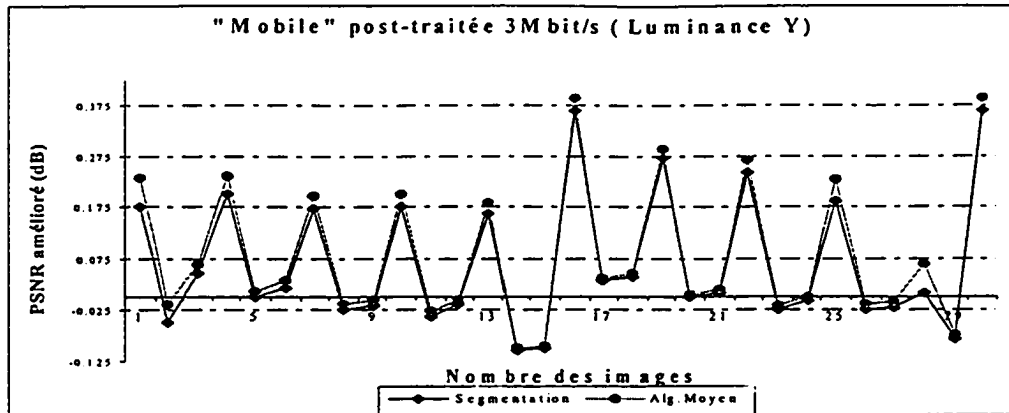
Figure 3.20 : Rapport signal sur bruit des séquences vidéo décodées

« Mobile » et « Flower » à 3 Mbit/s

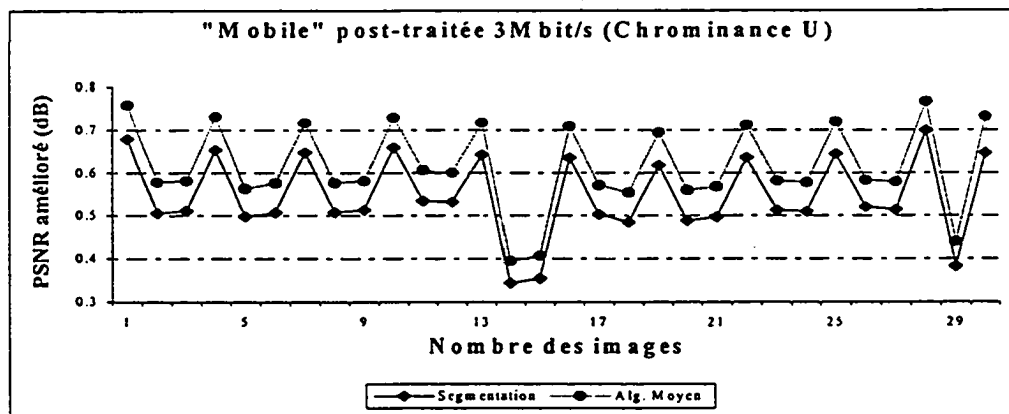
a) Luminance Y

b) Chrominance U

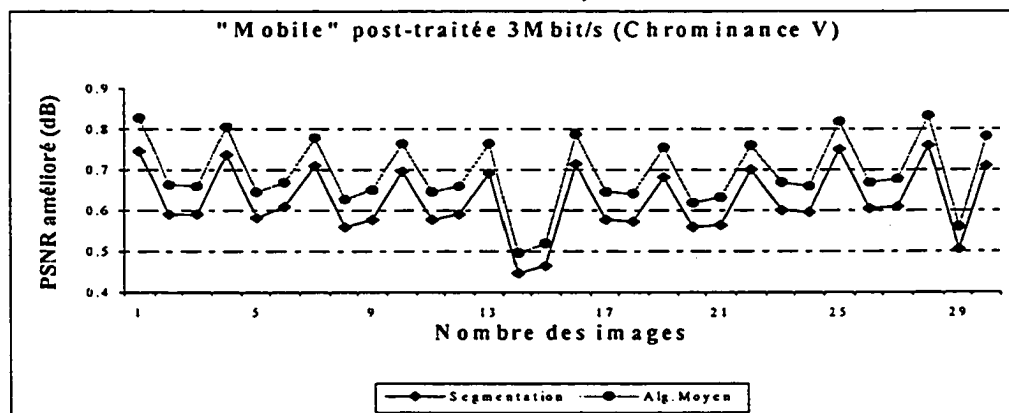
c) Chrominance V



a)



b)



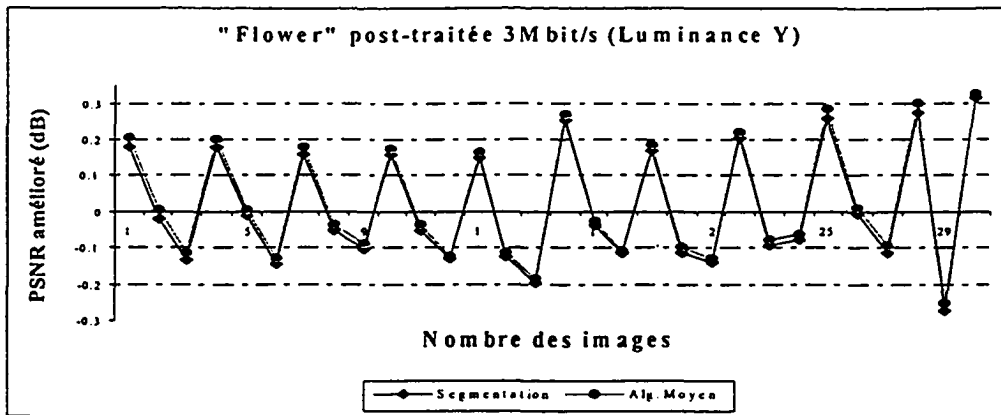
c)

Figure 3.21 : Rapport signal sur bruit du post-traitement pour la séquence « Mobile » à 3 Mbit/s traitée par l'utilisation de l'approche de segmentation et approche moyenne.

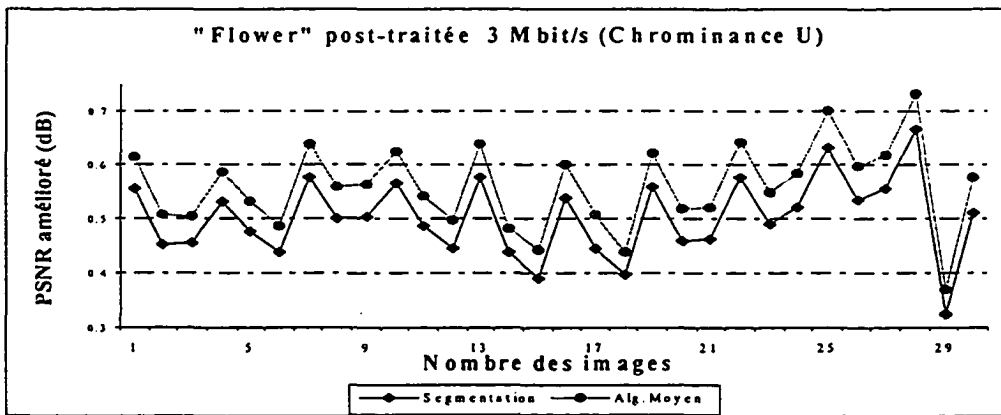
a) Luminance

Y b) Chrominance U

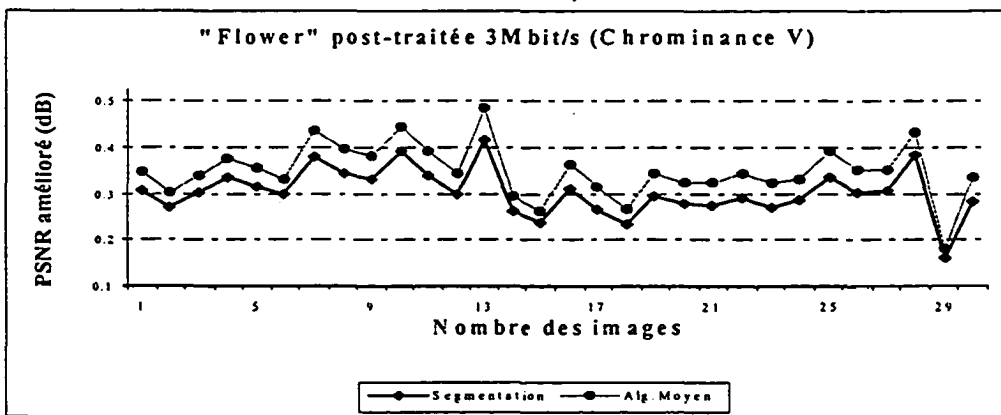
c) Chrominance



a)



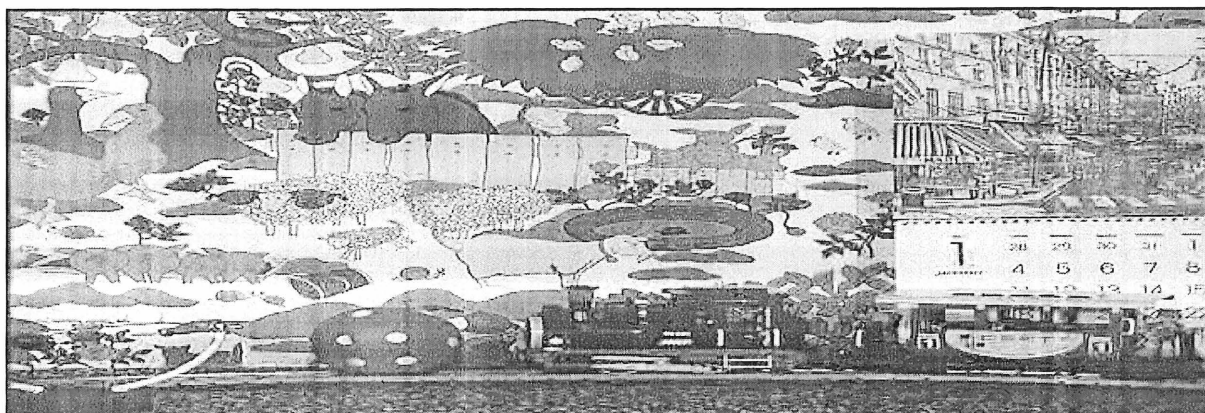
b)



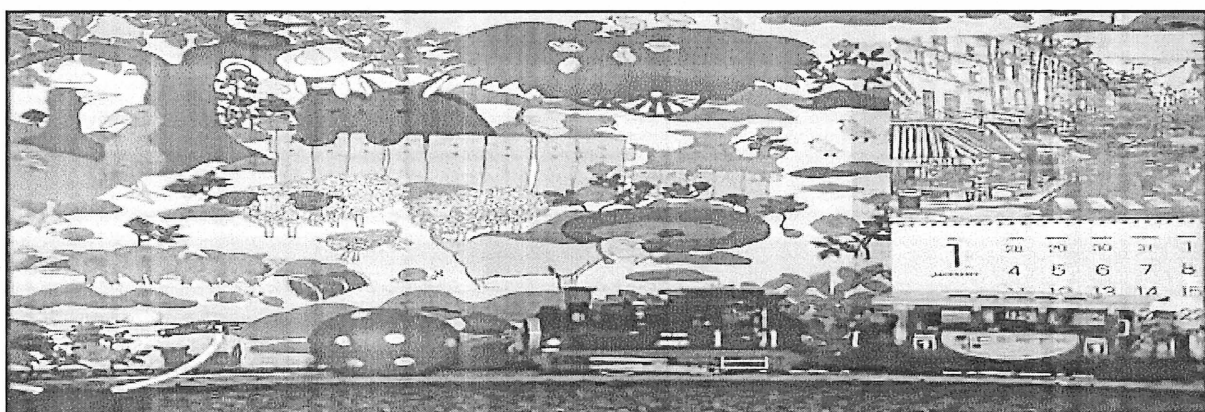
c)

Figure 3.22 : Rapport signal sur bruit du post-traitement pour la séquence « Mobile » à 3 Mbit/s traitée par l'utilisation de l'approche de segmentation et approche moyenne.

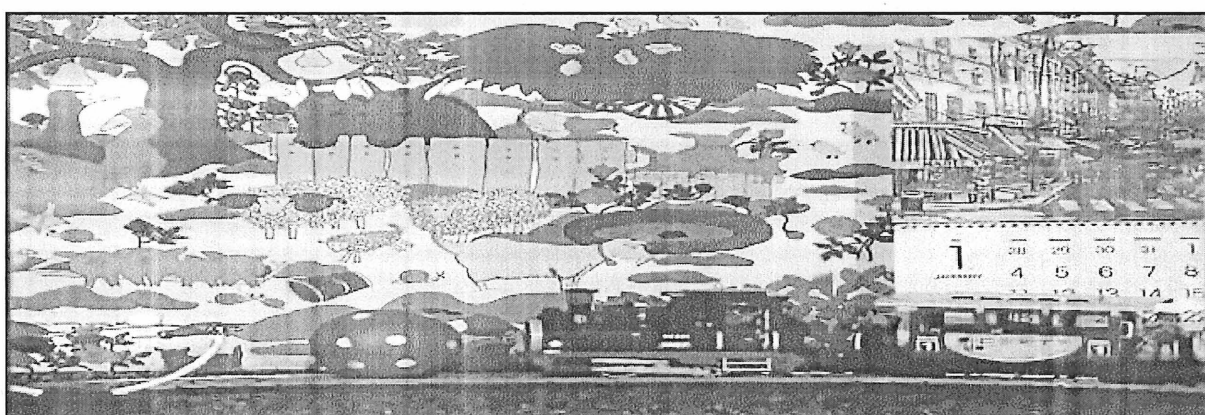
a) Luminance Y b) Chrominance U c) Chrominance



a)



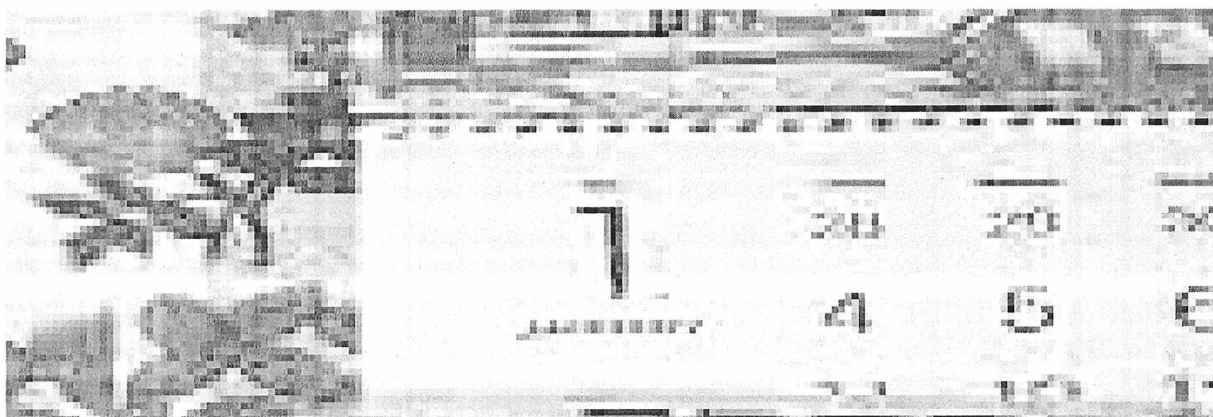
b)



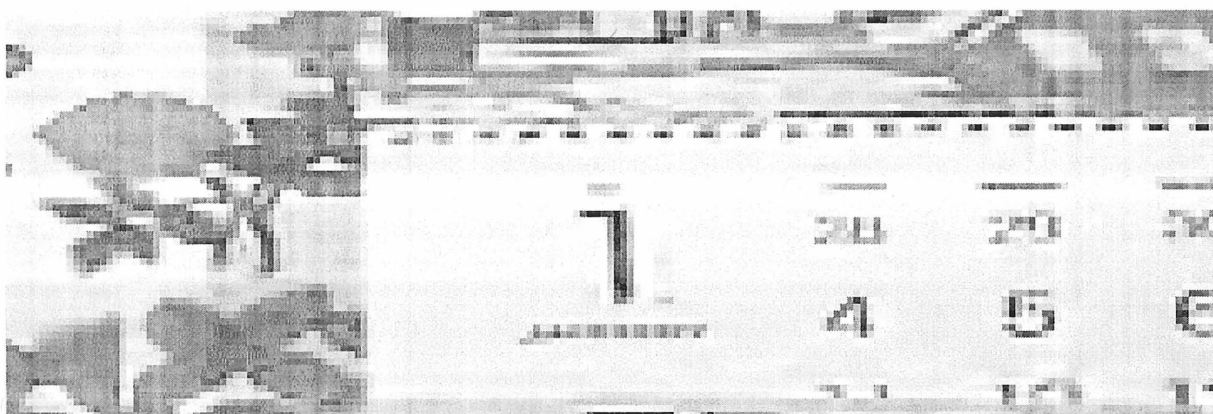
c)

Figure 3.23 : Résultats du post-traitement de la luminance Y pour « Mobile » à 3Mbit/s

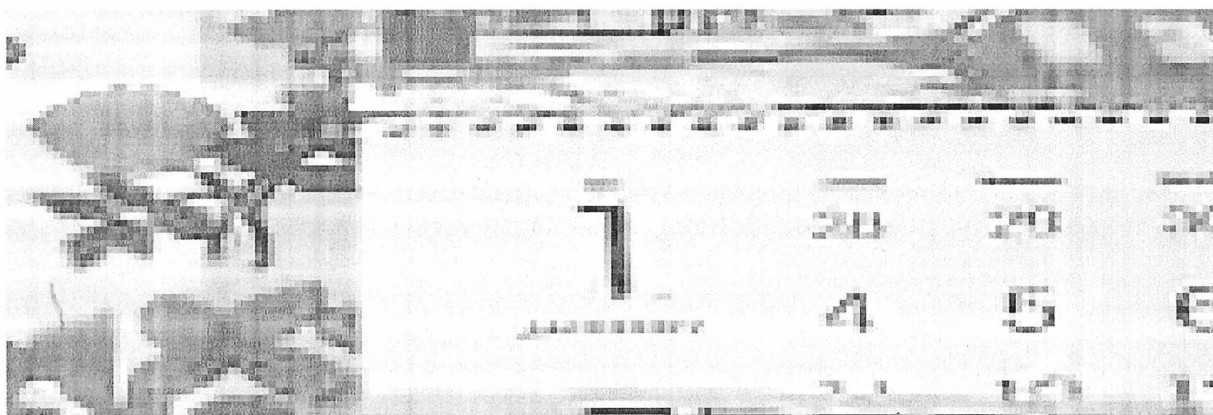
- a) Image décodée b) Image traitée par l'approche de segmentation
c) Image traitée par l'approche moyenne



d)



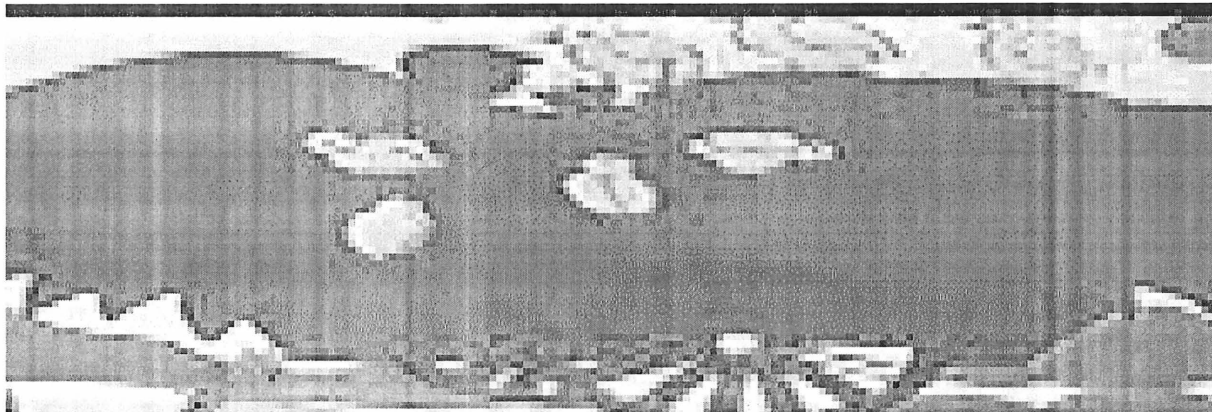
e)



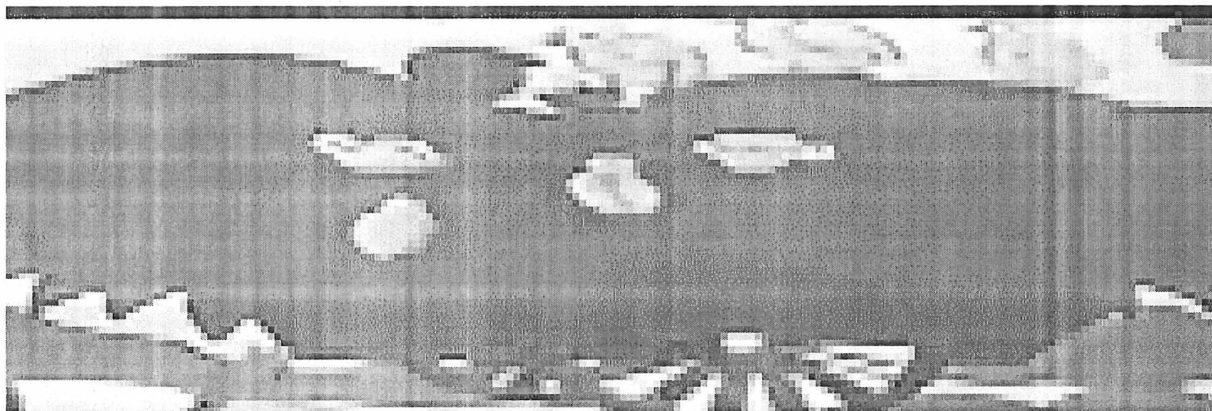
f)

Suite (Agrandissement)

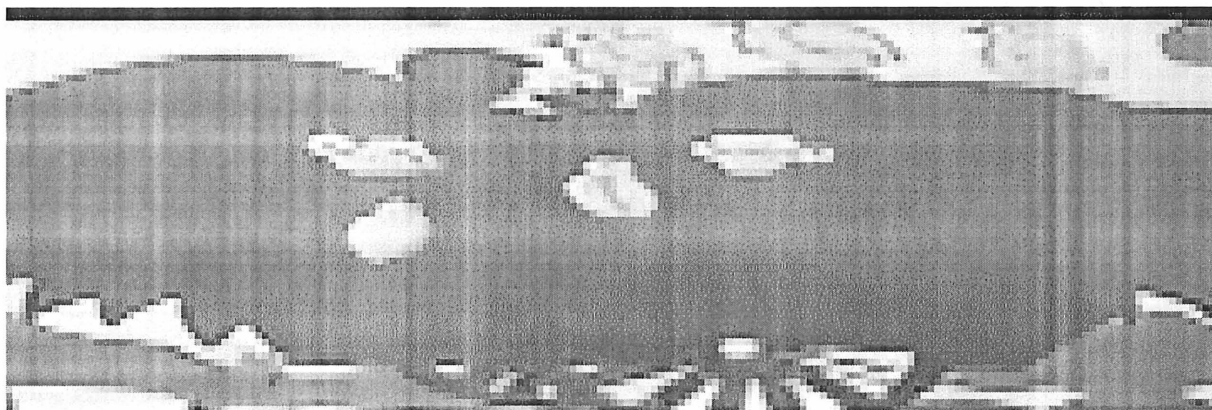
- d) Image décodée e) Image traitée par l'approche de segmentation
f) Image traitée par l'approche moyenne



g)



h)



i)

Suite (Agrandissement)

- g) Image décodée h) Image traitée par l'approche de segmentation
i) Image traitée par l'approche moyenne

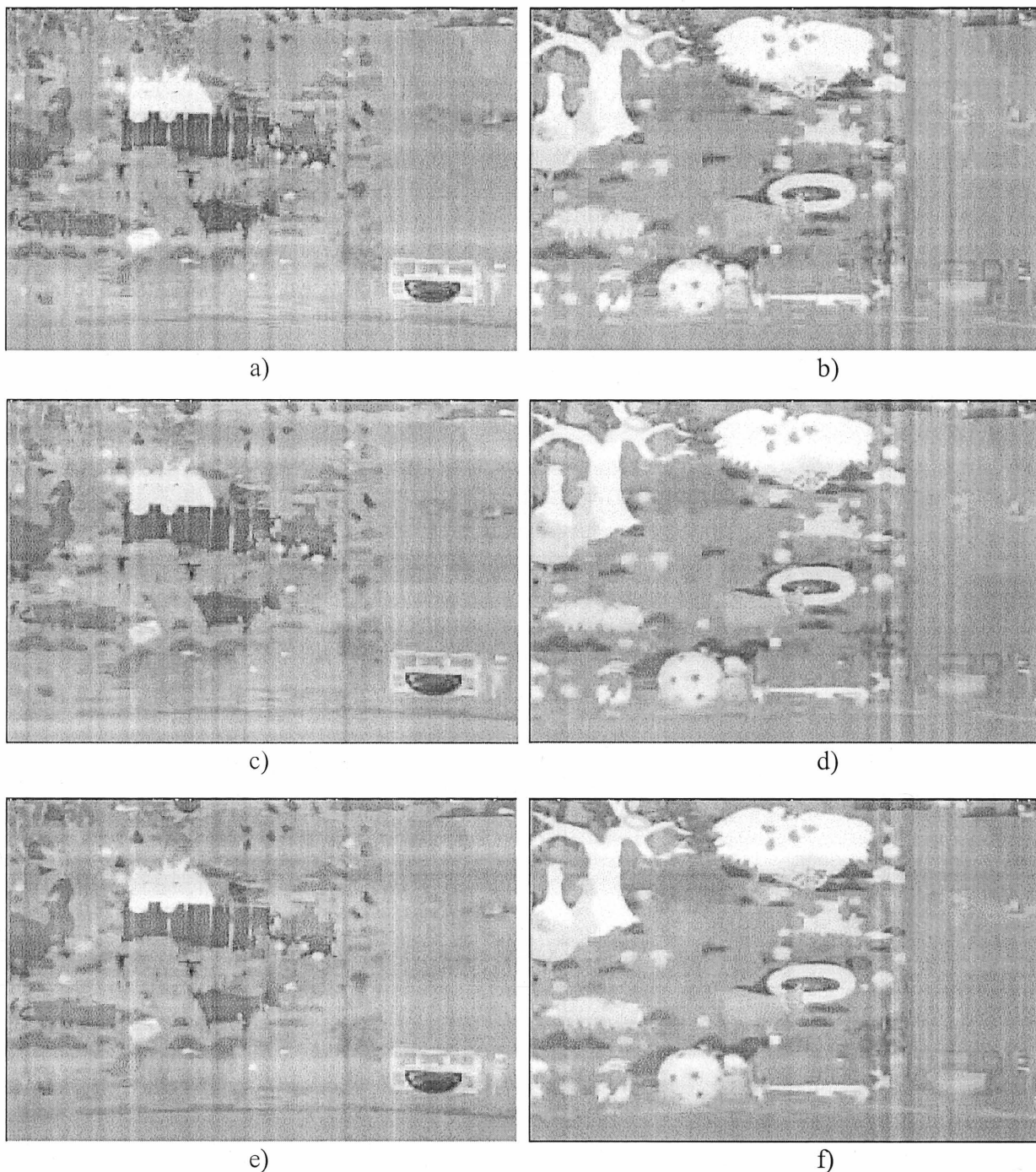
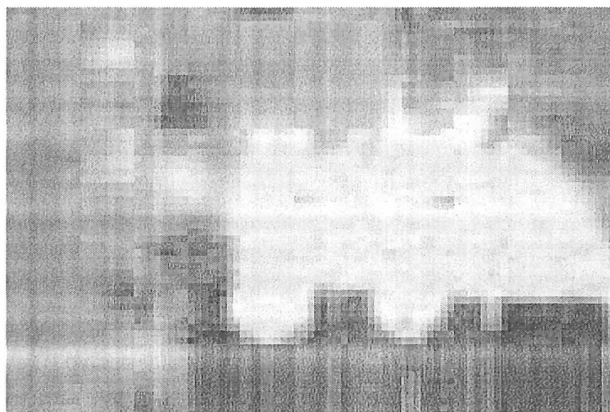
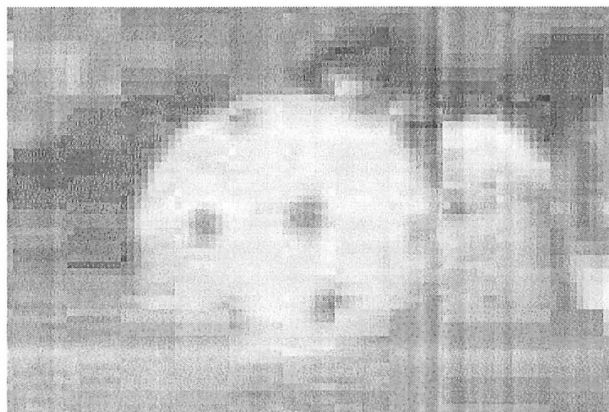


Figure 3.24 : Résultats du post-traitement des chrominance U et V pour « Mobile » à 3 Mbit/s

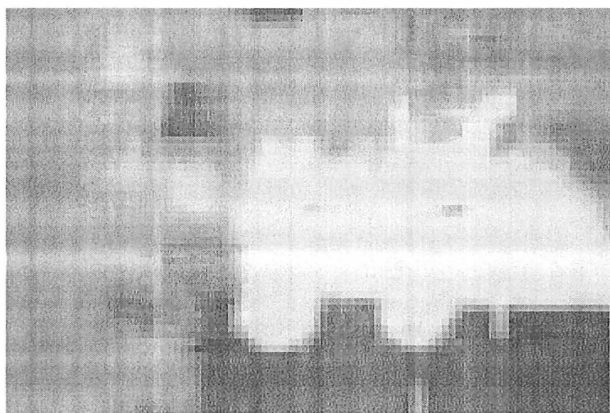
a) Image décodée U b) Image décodée V c) Image traitée U par l'approche de segmentation d) Image traitée V par l'approche de segmentation e) Image traitée U par l'approche moyenne f) Image traitée V par l'approche moyenne.



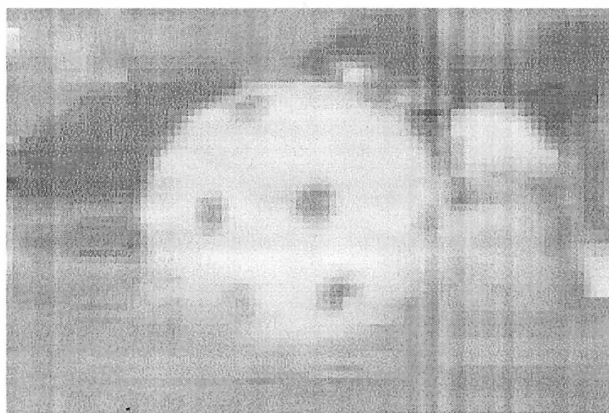
g)



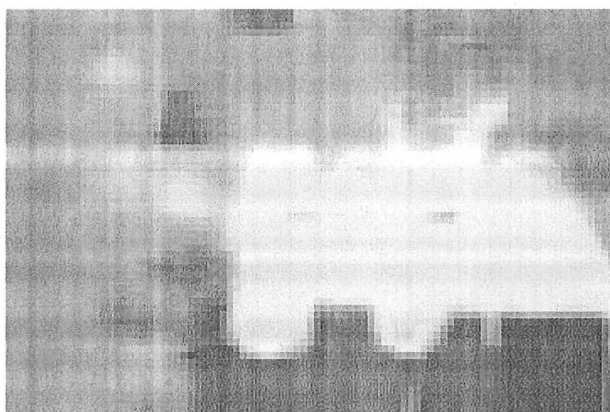
h)



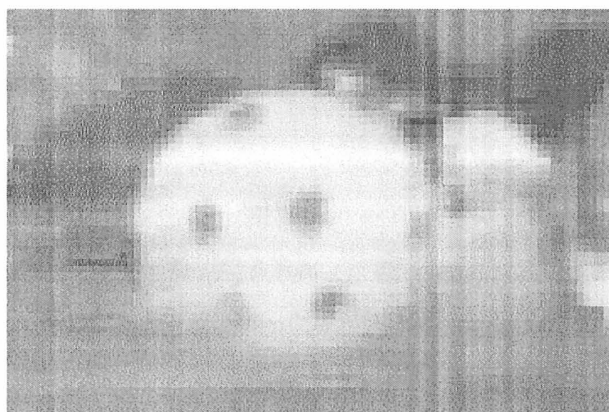
i)



j)



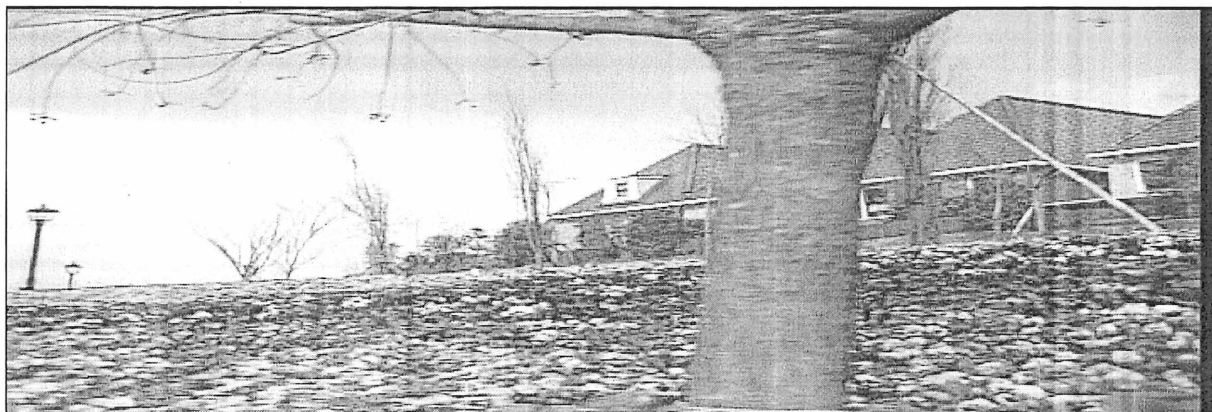
k)



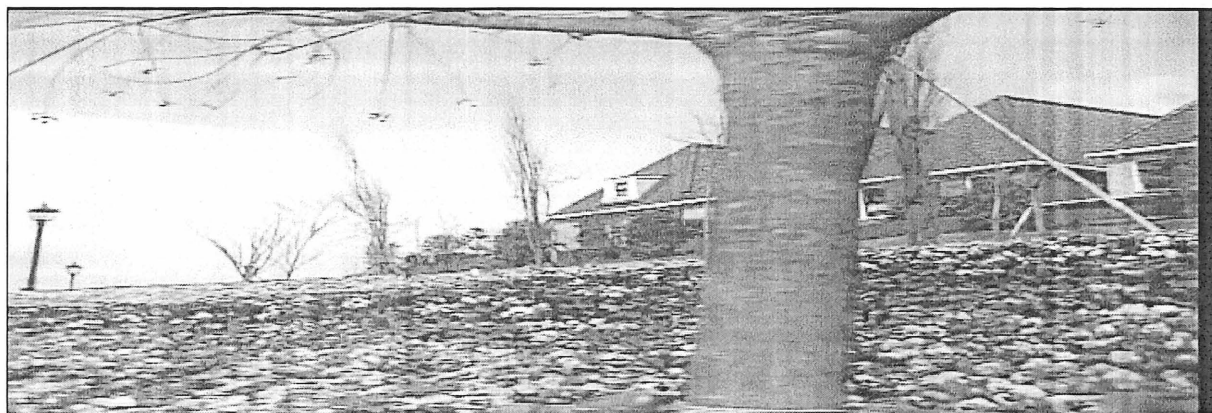
l)

Suite (Agrandissement)

g) Image décodée U h) Image décodée V i) Image traitée U par l'approche de segmentation j) Image traitée V par l'approche de segmentation k) Image traitée U par l'approche moyenne l) Image traitée V par l'approche moyenne.



a)



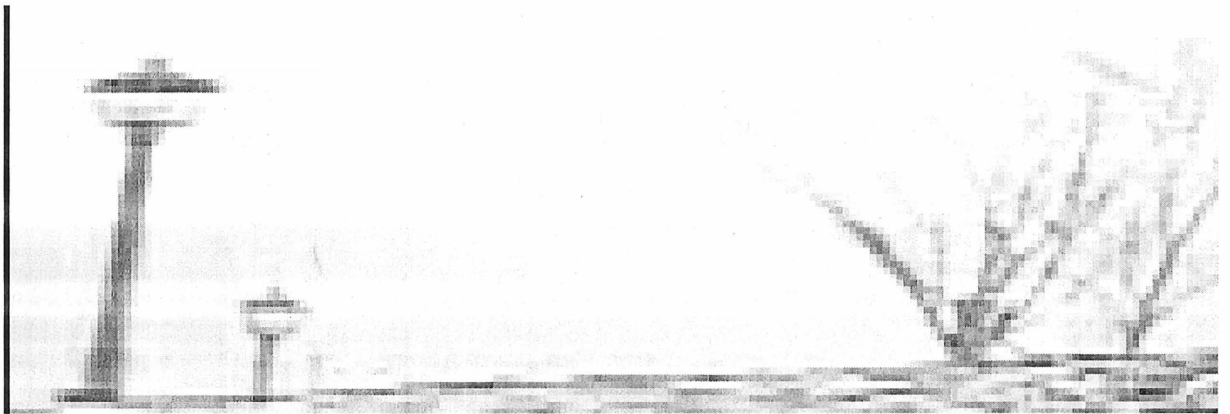
b)



c)

Figure 3.25 : Résultats du post-traitement de la luminance Y pour « Flower » à 3 Mbit/s

- a) Image décodée b) Image traitée par l'approche de segmentation
c) Image traitée par l'approche moyenne



d)



e)



f)

Suite (Agrandissement)

- d) Image décodée e) Image traitée par l'approche de segmentation
f) Image traitée par l'approche moyenne

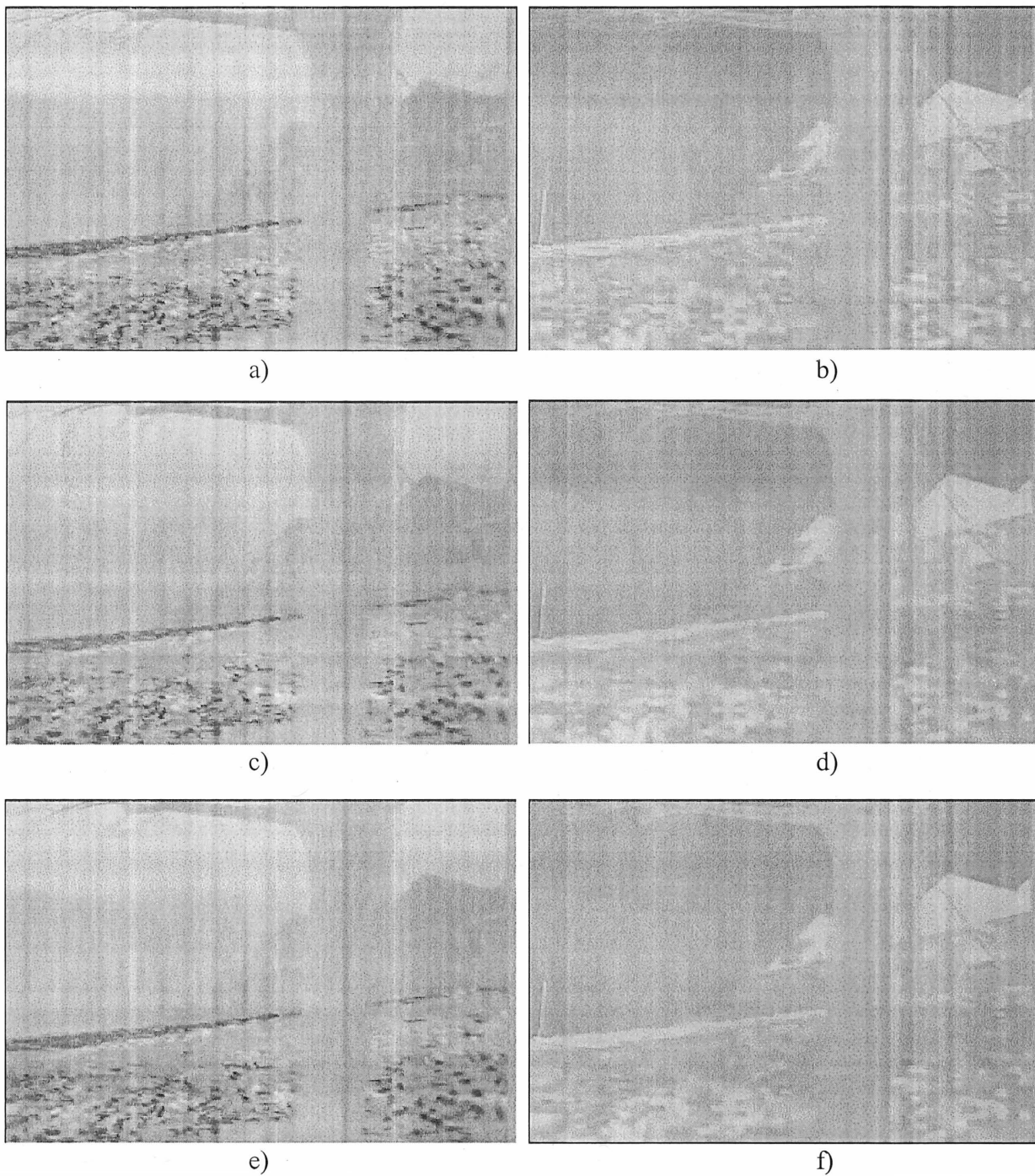
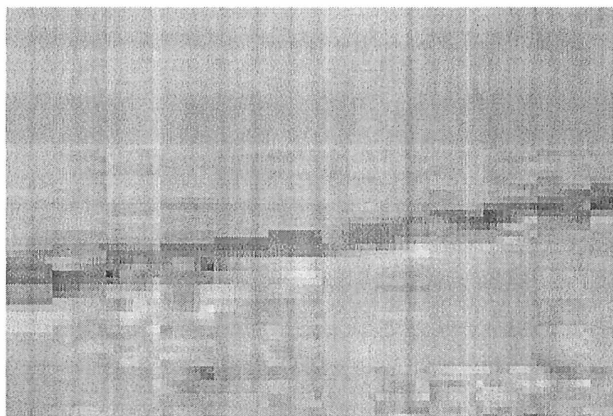
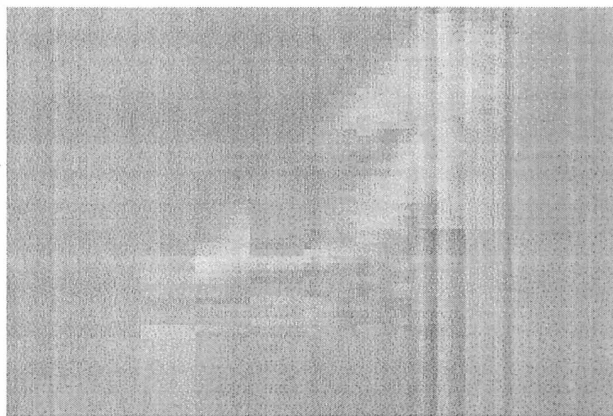


Figure 3.26 : Résultats du post-traitement des chrominance U et V
pour « Flower » à 3 Mbit/s

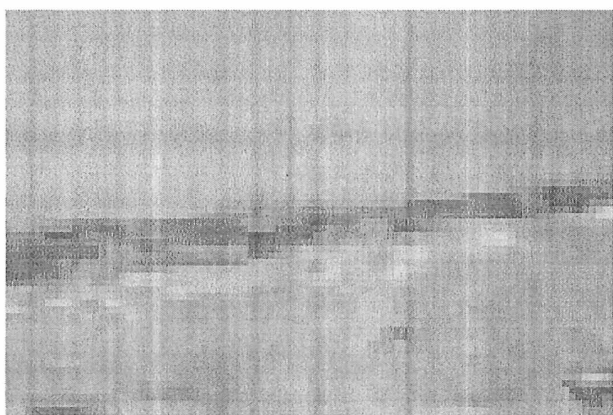
a) Image décodée U b) Image décodée V c) Image traitée U par l'approche de
segmentation d) Image traitée V par l'approche de segmentation e) Image traitée U
par l'approche moyenne f) Image traitée V par l'approche moyenne.



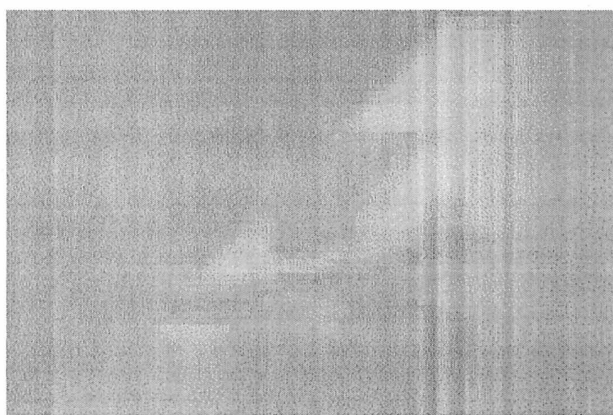
g)



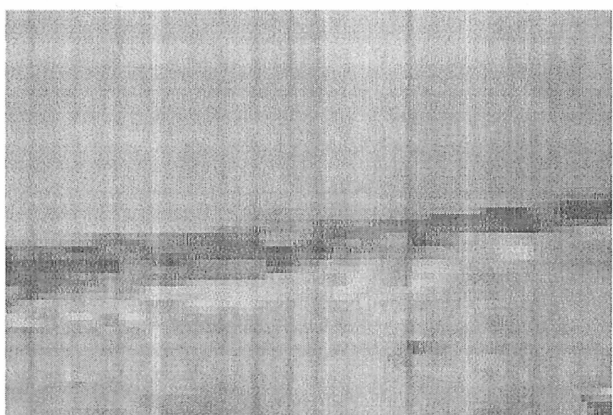
h)



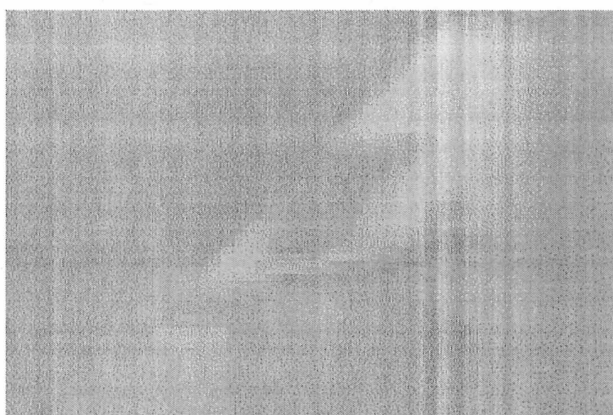
i)



j)



k)



l)

Suite (Agrandissement)

g) Image décodée U h) Image décodée V i) Image traitée U par l'approche de segmentation j) Image traitée V par l'approche de segmentation k) Image traitée U par l'approche moyenne l) Image traitée V par l'approche moyenne.

Conclusion

La présente recherche propose deux aspects reliés au traitement d'images. Le premier est la réduction du bruit gaussien additif dans les images. L'objectif principal de cette partie consiste à concevoir de nouveaux réducteurs de bruit adaptatifs qui permettent une meilleure performance que les réducteurs de bruit existants et à faire une comparaison détaillée entre eux. Le second est le post-traitement pour le standard de compression MPEG-2. Celui-ci vise à améliorer la qualité visuelle des images décodées avec un débit entre 3 Mbit/s et 8 Mbit/s.

Jusqu'à présent, plusieurs techniques spatiales de réduction de bruit additif ont été proposées. Mais leur performance est très variable. En général, chaque algorithme est efficace pour un type d'image seulement et il dépend beaucoup du niveau de bruit ajouté. Pour la réduction de bruit, nous avons proposé quatre nouveaux algorithmes adaptatifs spatiaux : un algorithme de segmentation, une approche utilisant la moyenne des images, une approche utilisant une image constante et une approche combinée qui exploite les avantages de ces deux dernières approches.

L'algorithme de segmentation est fondé sur le modèle des statistiques locales de Lee et sur la segmentation locale qui permet de classer le pixel considéré. Grâce à cette segmentation, l'utilisation du filtre spatialement variant devient plus précise et plus efficace. Cet algorithme donne une très bonne performance en comparaison avec les autres algorithmes existants. Il est supérieur pour toutes les images testées et tous les

niveaux de bruits. Quantitativement, son rapport signal sur bruit est plus grand que celui des trois autres de 0.5 dB jusqu'à 2 dB. Qualitativement, les images traitées sont plus naturelles et elles ne contiennent aucun artefact secondaire visible.

Les trois nouvelles approches (moyenne, constante et combinée) sont basées sur l'algorithme de segmentation. Selon l'algorithme de segmentation, la dimension de la fenêtre de traitement et la valeur de seuil S utilisée pour faire la segmentation sont tenues fixes, tandis que les structures sur les images et les niveaux de bruits changent d'une image à une autre. Alors, afin de mieux s'adapter aux caractéristiques de l'image contaminée et du bruit, ces nouvelles approches utilisent plusieurs fenêtres avec des valeurs de seuils différentes. Naturellement, la complexité de ces approches devient plus grande mais leur performance est considérablement augmentée. Au niveau du rapport signal sur bruit, leurs résultats sont entre 0.2 dB et 2.5 dB supérieurs à celui de l'algorithme de segmentation, suivant l'image à traiter et le niveau de bruit ajouté. Les images traitées par ces approches présentent aussi une meilleure qualité visuelle. Elles sont nettes et plus lisses dans les zones plates en comparaison avec les autres approches.

Concernant le post-traitement, nous avons développé un système composé d'un filtre en losange, d'un estimateur de bruit, d'un réducteur du bruit de moustique et du bruit de quantification, et d'un réducteur de l'effet de bloc. L'utilisation du filtre en losange a pour objectif d'estimer la puissance de bruit dans l'image et de réduire partiellement l'effet de bloc et le bruit aléatoire. L'estimateur de bruit utilise l'image filtrée par le filtre en losange et une fonction déterminée de façon empirique pour calculer la variance du bruit contenu dans chaque image de la séquence vidéo. En se basant sur cette information estimée et sur l'un des deux algorithmes : algorithme de segmentation ou l'approche moyenne, le réducteur du bruit moustique et du bruit de quantification est conçu. La segmentation utilisée dans ces algorithmes permet d'enlever la majeure partie du bruit de moustique parce que dans les images MPEG-2, la plupart de ce bruit se trouve dans les zones autour des contours. Finalement, le réducteur d'effet de bloc est utilisé pour enlever ces défauts de l'image. Puisque la

localisation de l'effet de bloc est bien connue, nous divisons l'image à améliorer en quatre bandes : basse-basse, basse-haute, haute-basse et haute-haute. Les trois sous-images à haute fréquence, qui présentent l'effet de bloc sur les directions verticales, horizontales et diagonales, sont filtrées spatialement, toujours en utilisant les principes des statistiques locales et de la segmentation.

Avec notre système de post-traitement, nous ne gagnons pas beaucoup en matière du rapport signal sur bruit (environ 0.05 dB pour la luminance Y et environ 0.5 dB pour les chrominances U et V), mais les images traitées garantissent une grande amélioration de la qualité visuelle. Elles contiennent beaucoup moins de défauts et, surtout dans les zones en mouvement, l'effet de papillotement est fortement réduit. En fait, notre système de post-traitement peut être utilisé aussi pour améliorer la qualité des images fixes compressées par le standard JPEG.

Bibliographie

- [1] BANHAM M.R., KATSAGGELOS A. K. (1996) *Spatially Adaptive Wavelet-Based Multiscale Image Restoration*, IEEE Transactions on Image Processing, Vol. 5, No. 4, pp. 619-634.
- [2] BHASKARAN V., KONSTANTINIDES K. (1995) *Image and Video Compression Standards: Algorithms and Architectures*, Kluwer Academic Publisher.
- [3] CHAN P., LIM J. S. (Février 1985) *One-Dimensional Processing for Adaptive Image Restoration*, IEEE Transaction on Acoustics, Speech, and Signal Processing, Vol. ASSP-33, No. 1, pp. 117-126.
- [4] DONOHO D. L., JOHNSTONE I. M. (1995) *Adapting to Unknown Smoothness via Wavelet Shrinkage*, J. Amer. Stat. Assoc., Vol. 90.
- [5] GONZALEZ R. C., WOODS R. E. (1993) *Digital Image Processing*, Addison-Wesley Publishing Company.
- [6] HUANG T. S. (1979) *Picture Processing and Digital Filtering*, New York, Springer-Verlag.
- [7] HUANG T. S. (1981) *Two Dimensional Digital Signal Processing II: Transform and Median Filters*, New York, Springer-Verlag.
- [8] KIEU, CONG-TOAI (1997) *Pré-et post-traitement pour le signal vidéo MPEG-1*, Thèse de maîtrise, Département de génie électrique et de génie informatique, Université de Sherbrooke.
- [9] KUAN D. T, SAWCHUK A. A. (March 1985) *Adaptive Noise Smoothing Filter for Images with Signal-Dependent Noise*, IEEE Transaction on Pattern Analysis and Machine Intelligence, Vol. PAMI-7, No. 2, pp. 165-77.
- [10] KUO C. J., HSIEH R. J. (1995) *Adaptive Postprocessor for Block Encoded Images*, IEEE Transactions on Circuits and Systems for Video Technology, Vol. 5, No. 4, pp. 298-304.

- [11] LEDINH CHON-TAM. (Automne 1996) *Traitement des images*, Cours gradué GEI754, Département de génie électrique et de génie informatique, Université de Sherbrooke.
- [12] LEDINH CHON-TAM, KIEU CONG-TOAI, DO VIET-HA (1999) *Apparatus and Methods for Segmentation Based Adaptive Spatial Noise Reducer*, USA Patent Application.
- [13] LEE JONG-SEN. (Mars 1980) *Digital Image Enhancement and Noise Filtering by Use of Local Statistics*, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. PAMI-2, No. 2, pp. 165-168.
- [14] LEE JONG-SEN (1983) *Digital Image Smoothing and Sigma Filter*, Computer Vision, Graphics, and Image Processing, No. 24, pp. 255-269.
- [15] LEGALL D. (1991) *MPEG : A Video Compression Standard for Multimedia Applications*, Communication of the ACM, Vol. 34, No. 4, pp. 47-58.
- [16] LIM. J. S (1990) *Two-Dimensional Signal and Image Processing*, New Jersey, Prentice Hall PTR, Englewood Cliffs.
- [17] LIU T., JAYANT N. (1995) *Adaptive Postprocessing Algorithms for Low Bit Rate Video Signals*, IEEE Transactions on Image Processing, Vol. 4, No. 7, pp. 1032-1035
- [18] LUO J., CHEN C., PARKER K., HUANG T. (1994) *A New Method for Block Effect Removal in Low Bit Rate Image Compression*, Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, Vol. 5, pp. 341-344.
- [19] MALFAIT M., ROOSE D. (1997) *Wavelet-Based Image Denoising Using a Markov Random Field a Priori Model*, IEEE Transactions on Image Processing, Vol. 6, No. 4, pp. 549-565.
- [20] MCDONNALL M J. (1981) *Box-Filtering Techniques*, Computer Graphics Image Processing, No. 17, pp. 65-70.
- [21] NAGAO M., MATSUYAMA T. (1979) *Edge Preserving Smoothing*, Computer Graphics Image Processing, No. 9, pp. 394-407.
- [22] PAEK H., PARK J. W., LEE S. U. (1995) *Non-Iterative Postprocessing Technique for Transform Coded Image Sequence*, IEEE Transactions on Image Processing, pp. 208-211.

- [23] RANK K., UNBEHAUEN R. (1992) *An Adaptive Recursive 2-D Filter for Removal of Gaussian Noise in Images*, IEEE Transactions on Image Processing, Vol. 1, No. 3, pp. 431-436.
- [24] PARK J., KURZ L. (May 1996) *Image Enhancement Using the Modified ICM Method*, IEEE Transactions on Image Processing, Vol. 5, No. 5, pp. 765-771.
- [25] PODILCHUK C., SANFRANEK R. J. (1997) *Image and Video Compression: A Review*, International Journal of High Speech Electronics and Systems, Vol. 8, No. 1, pp. 119-177.
- [26] PRATT W. K. (1991) *Digital Image Processing*, New York, 2nd édition, Wiley.
- [27] RAMAMURTHI B., GERSHO A. (1986) *Nonlinear Space-Variant Postprocessing of Block Coded Images*, IEEE Transactions on Acoustics, Speech and Signal Processing ASSP-34, pp. 1258-1268.
- [28] REEVES H. C., LIM J. S. (1984) *Reduction of Blocking Affects in Image Coding*, Optical Engineering, Vol. 23, No. 1, pp. 34-37.
- [29] SCHWARTZ M., SHAW L. (1975) *Signal Processing: Discrete Spectral Analysis, Detection, and Estimation*, McGraw-Hill Inc.
- [30] SHU Y. F., CHEN Y. C. (1993) *A New Separable Median Filter for Removing Blocking affects*, IEEE Transactions on Consumer Electronics, Vol. 39, No. 3, pp. 510-513.
- [31] YANG S., HU Y. H. (1997) *Coding Artifacts Removal Using Biased Anisotropic Diffusion*, IEEE International Conference on Image Processing, Vol. II, pp. 346-349.
- [32] YANG Y, GALATSANOS N., KATSAGGELOS A. (1993) *Iterative projection algorithms for removing the blocking artifacts of block-DCT compressed images*, Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, Vol. 5, pp. 405-408.
- [33] YANG Y, GALATSANOS N., KATSAGGELOS A. (1993) *Regularized Reconstruction to Reduce Blocking Artifacts of Block Discrete Cosine Transform Compressed Images*, IEEE Transactions on Circuits and Systems for Video Technology, Vol. 3, pp. 421-432.
- [34] YASUYUKI N., HIRONAO H., TAMOTSU K. (1994) *A Pel Adaptive Reduction of Coding Artifacts for MPEG Video Signals*, ICIP, Texas, pp. 928-932.

- [35] ZAKHOR A. (1992) *Iterative Procedures for Reduction of Blocking Effects in Transform Image Coding*, IEEE Transactions on Circuits and Systems for Video Technology, Vol. 2, No. 1, pp. 91-95.